AD-A045 081 FLORIDA UNIV GAINESVILLE DEPT OF INDUSTRIAL AND SYS--ETC F/G 9/2
AN INTERACTIVE HEURISTIC APPROACH FOR SCHEDULING A MULTI RESOUR--ETC(U)
SEP 77 P CARLSON, T J HODGSON N00014-76-C-0096

UNCLASSIFIED RB-77-10 NL

1 OF 1
ADA045081

END
DATE
FILMED
11-77
DDC

10

2

RESEARCH REPORT

Industrial & Systems
Engineering Department
University of Florida
Gainesville, FL. 32611

10

AN INTERACTIVE HEURISTIC APPROACH FOR SCHEDULING A MULTI RESOURCE CONSTRAINED SYSTEM

Research Report No. 77-10

by

PER-OLOF CARLSON
THOM J. HODGSON

September, 1977

Department of Industrial and Systems Engineering
University of Florida
Gainesville, Florida 32611

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

This research was supported in part by the Office of Naval Research, under contract number N00014-76-C-0096.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
77-10

2. GOVT ACCESSION NO.

3. ~~RECIPIENT'S CATALOG NUMBER~~
9 Research rept.

4. TITLE (*and Subtitle*)
6 An Interactive Heuristic Approach for Scheduling a Multi Resource Constrained System.

5. ~~TYPE OF REPORT & PERIOD COVERED~~

6. ~~PERFORMING~~ ORG. REPORT NUMBER
14 RR-77-10

7. AUTHOR(s)
10 Per-Olof Carlson
Thom J. Hodgson

8. ~~CONTRACT~~ OR GRANT NUMBER(s)
15 N00014-76-C-0096

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Industrial & Systems Engineering
University of Florida
Gainesville, FL 32611

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research
Arlintgon, VA

12. ~~REPORT DATE~~
11 September, 1977

13. ~~NUMBER OF PAGES~~
69
12 69p.

14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)

15. SECURITY CLASS. (*of this report*)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

N/A

404 399

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Scheduling
Resource Constraint
Interactive

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

The solution to a multi resource, multi project scheduling problem is approached in a way that combines human interaction, heuristics and optimality seeking procedures. A tree search algorithm is employed to perform the search. By use of heuristics, the tree can be "pruned" so as to limit its branches, i.e., the problem is simplified and computational times for optimality seeking procedures can be drastically reduced. By man-computer

Next Page →

20.

interaction, data and constraints can be changed and the search through the tree controlled. The model developed has been implemented using a minicomputer and a video display. To help the operator control the scheduling procedure different graphical displays are available. The system is characterized by being flexible to changes in data and constraints, and to allow the operator to direct the computer in the search for a solution to the scheduling problem.

TABLE OF CONTENTS

Page

FORWARD

This report details the development of an experimental interactive scheduling system. The system has been implemented on a PDP-11/20 with a CRT and hard copy terminal. The system has been designed with a maximum of flexibility to facilitate future research into the most effective combinations of man/machine interaction. As such, this report should be considered to be Part One of an ongoing effort.

ABSTRACT

The solution to a multi resource, multi project scheduling problem is approached in a way that combines human interaction, heuristics and optimality seeking procedures. A tree search algorithm is employed to perform the search. By use of heuristics the tree can be "pruned" so as to limit its branches, i.e., the problem is simplified and computational times for optimality seeking procedures can be drastically reduced. By man-computer interaction, data and constraints can be changed and the search through the tree controlled. The model developed has been implemented using a minicomputer and a video display. To help the operator control the scheduling procedure different graphical displays are available. The system is characterized by being flexible to changes in data and constraints, and to allow the operator to direct the computer in the search for a solution to the scheduling problem.

CHAPTER I

INTRODUCTION

1.1 A Scheduling Problem

Consider the situation when there is a pool of various resources, which are to be utilized by different projects over a given time period or planning horizon. Each of the projects wants to use a specific configuration of the resources for a certain amount of time. To satisfy the projects' requests it is desirable to sequence them in such a manner that the available resources are used in an efficient way. To accomplish this it is necessary to find a schedule for the specified planning horizon (e.g., next month, quarter, year) telling when different projects may employ the resources and how the utilization of the various resources changes over time. This schedule (however arrived at) will be called an initial schedule.

In creating an initial schedule certain conditions are given (e.g., the number of projects and their requests for resources), that in real world situations are subject to change. The initial schedule may become out-of-date and it would be necessary to update it. As examples consider the following three situations:

(1) A new high priority project needs to be fit into the schedule;

(2) Some of the resources may break down and not be available for use during a certain time period(s); or

(3) New information may be obtained relative to future changes in availability of resources (e.g., planned maintenance, addition of new resources).

The need to update the initial schedule can occur at any time. Once the initial schedule is updated, it could then become necessary to change the updated schedule.

To handle scheduling problems of this type in practical applications it should be advantageous to use a computer aided scheduling system. Indeed, this thesis is concerned with the development of an interactive computerized system for problems of this type. The objective is to build a system that will allow the operator to explore the area between doing the scheduling by hand and letting the computer do the scheduling. The stated problem generally is too complex for the human being to solve by hand. The computer also will have problems evaluating the extremely large number of potential schedules for problems of this type without some enlightened guidance by the operator. Therefore, a man-computer interactive system will be developed, which allows the operator to guide the computer towards finding an acceptable (if not optimal) schedule.

1.2 Related Work

The scheduling problem at hand in this study is similar to the classical resource constrained critical path method (CPM) problem. Since the introduction of the CPM-technique in the 1950's many publications have appeared dealing with various aspects of the CPM-scheduling problem. For the interested reader, excellent surveys of the developments in this area have been published by Davies [5] and Herroelen [9]. The original CPM models focus on time and are based on the implicit assumption that unlimited resources are available. The schedules developed from these models imply specific profiles of resource usage over time. There is no guarantee that resource usage will not exceed available levels. While the assumption of unlimited resources may be justified in some cases, many schedulers are faced with the problem of scheduling with limited resources.

The resource constrained CPM problem is representative of a class of combinatorial problems. They all have the same general form and are characterized by an exponential growth in the potential number of solutions as the problem size increases. Besides the resource constrained CPM problem there are related scheduling problems such as the job-shop scheduling problem [4,13,20] and the assembly line balancing problem [12,23]. With regard to the type of results produced in attempting to solve these types of problems, the solution procedures may be grouped into two major categories. First, there are heuristic procedures that aim at producing good, feasible schedules. These procedures involve the use of some rule-of-thumb or heuristic to determine priorities among jobs. Second, there are procedures which aim at producing the best possible, or optimal, schedule. To date experience with problems of the type discussed here indicates that optimal solution procedures are for all practical purposes computationally unfeasible. Solution procedures have been proposed using integer programming [3,8,16,17,18] and enumeration techniques [2,6,10,19]. The state of the art for these techniques (except for certain specially structured problems) is insufficiently advanced to solve efficiently scheduling problems of the type considered here. As a consequence, solutions which have been operationally useful have been based primarily on heuristic procedures.

One of the earliest and more ambitious heuristic procedures was developed by J. D. Wiest. His SPAR (Scheduling Program for Allocating Resources) program [22] is a heuristic model for scheduling large projects. In its basic approach, the model focuses on available resources which it serially allocates, period by period, to jobs listed in order of their early start times. Jobs are scheduled, starting with the first period, by selecting from the list of those currently available and ordered according to their total slack (which is based on technological constraints only and normal resource assignments). The most critical jobs have the highest probability of being scheduled first, and as many jobs are scheduled as available resources permits. If an available job fails to be scheduled in that period, an attempt is made to schedule it the next period. Eventually all jobs so postponed become critical and move to the top of the priority list of available jobs. The basic program is modified by a number of additional scheduling heuristics designed to increase the use of available resources and/or to decrease the length of the schedule.

Leahy [11] described the development of what was probably the first man-machine interactive CPM-system in which messages were presented on-line as resource conflicts arose and alternate resource allocations specified on-line. Furthermore a report by Paulson [14] lays out the specifications of a computer-based project control system utilizing on-line graphics and a man-machine interactive approach to resource scheduling.

In [21] J. D. Wiest describes the adaptation of his SPAR program to a man-machine interactive system for project scheduling in which computer graphics are employed for allocating resources to activities in a project network. The user can quickly generate feasible schedules and graphically display the results on a CRT (cathode ray tube). By means of a light pen and a table [21], he can adjust resource levels, change scheduling parameters or modify program heuristics, and readily determine the effects of these changes on the project schedule. Wiest points out that "the combination of man's intuitive skills and the computer's computational powers enhances the effectiveness of both in searching for good project schedules."

At Purdue University [1,15] an interactive graphics system for CPM scheduling has been developed. The system features include the ability to create, analyze, review, edit, save and retrieve alternative schedules. The system is implemented on a digital minicomputer PDP 11/40, and an Imlac PDS-1 intelligent display processor and a Calcomp 502 flatbed plotter. The CPM system is written in FORTRAN IV and the operating system performs all of the scheduling and diagram generation calculations. A FORTRAN IV dynamic graphics subroutine package in the PDP 11/40 generates graphic commands that are processed by the PDS-1 operating system to create the CPM arrow diagrams on the CRT. The operation of the system consists of first creating an input data file that describes the activities in the project. From this input file a CPM diagram is constructed. This diagram can be reviewed by "zooming" in to look closely at any part. By "pointing" to an activity on the CRT screen (with a device consisting of two perpendicularly mounted rotary potentiometers) the entire data base associated with it can be displayed on the screen. Any activity input variable can then be changed through graphic interaction and a revised diagram generated in a matter of seconds.

1.3 Scope

From above it is obvious that a system using only an optimality seeking procedure is computationally inappropriate. Therefore an approach that combines human interaction, heuristics and optimality seeking procedures will be undertaken. While the computer operates at a very high speed, it does not have man's capability of pattern recognition and intuitive analysis. The heuristics used in this study aim at simplifying the problem by limiting the domain of search to find a feasible schedule. A tree search algorithm will be employed to perform the search. By man-computer interaction the tree can be "pruned" so as to limit its branches. In the subsequent chapter the scheduling procedures and heuristics will be described. Chapter III outlines the whole scheduling system and gives some examples of schedules created. In the final chapter the system is critically assessed and some recommendations are given for future development.

CHAPTER II

SCHEDULING PROCEDURES AND HEURISTICS

2.1 Formulation of the Model

The following definitions are necessary:

r_{tj} = Available amount of resource j during time period t

q_{kji} = Required amount of resource j during the kth time period of the i^{th} project's duration

$X_{ti} = \begin{cases} 1, \text{ if project i is started during time period t} \\ 0, \text{ otherwise} \end{cases}$

d_i = Duration of project i

Z = Maximum overload of resources

If it is assumed that resource units are interchangeable (see p. 37) the problem to be addressed here can be formulated as follows

$$\text{Min } Z$$

$$(1) \quad \text{s.t.} \sum_{i} \sum_{k=1}^{d_i} q_{kji} \quad X_{t-k+1,\, i} - Z \leq r_{tj} \qquad \begin{array}{l} t = 1, \ldots, T \\ j = 1, \ldots, R \end{array}$$

$$(2) \quad \sum_{t} X_{ti} = 1 \qquad i = 1, \ldots, P$$

$$X_{ti} = 0, 1 \qquad \forall i, t$$

$$Z \geq 0$$

Constraint set (1) may be divided into two subsets. The first one (S_1) comprises all resource constraints that cannot be violated (hard constraints), i.e., no overload is permitted. The second subset (S_2) contains those resource constraints that may be violated (soft constraints). With respect to this, the problem may now be formulated as,

$$\text{Min } Z$$

$$(3)\quad \text{s.t.} \sum_{i} \sum_{k=1}^{d_i} q_{kji} \cdot X_{t-k+1,i} \le r_{tj} \qquad t,\ j \epsilon S_1$$

$$(4)\quad \sum_{i} \sum_{k=1}^{d_i} q_{kji} \cdot X_{t-k+1,i} - Z \le r_{tj} \qquad t,\ j \epsilon S_2$$

$$(5)\quad \sum_{t} X_{ti} = 1 \qquad i = 1, \ldots, P$$

$$X_{ti} = 0,\ 1 \qquad \forall t,\ i$$

$$Z \ge 0$$

The problem is to find X_{ti}'s such that the hard constraints are met and the maximum overload (Z) is minimized. This is a minimax 0-1 integer programming problem.

To find a solution to the minimax problem a tree search algorithm is used (see 2.3). The tree has one level for each project (see Figure 2.1). The root of the tree is designated as level 0. Each branch emanating from a node represents assigning a start time to that project. Assuming the search has reached a node at level i, a partial solution $X_{t,1}$, $X_{t,2}$, ..., $X_{t,i-1}$ is defined. At level p, a feasible schedule is found. The search proceeds downward and left-to-right at each level, backtracking when it is determined that

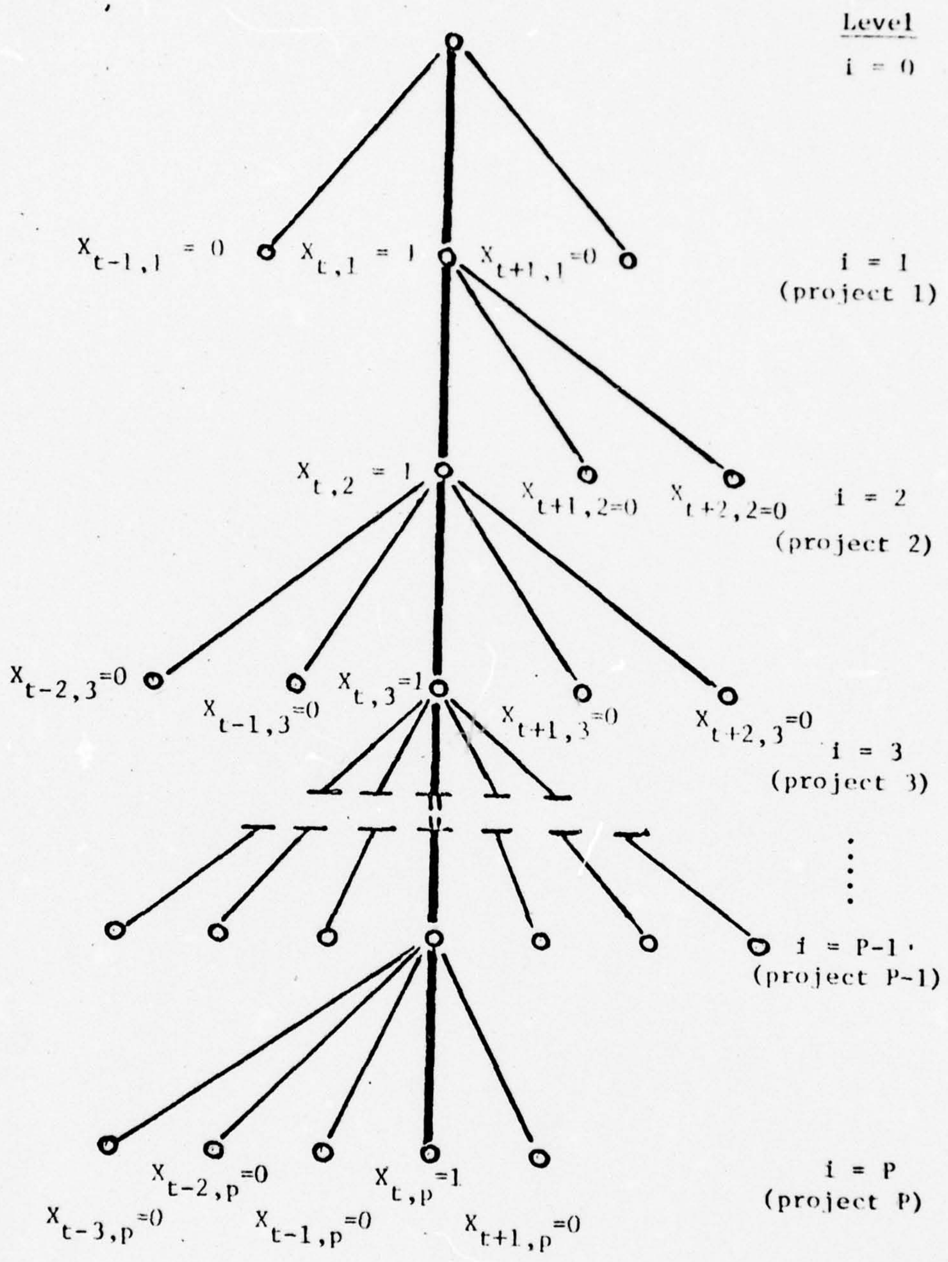


Figure 2.1. Tree Representation

(a) a terminal node has been reached, which implies that a feasible schedule has been found.

(b) further downward search will produce an unfeasible schedule.

The algorithm terminates when the search backtracks to the root of the tree (level = 0), when an optimal solution is found (Z = 0), or, when the execution time has exceeded a pre-set limit.

The algorithm employs a bounding function B_{tr} to determine if potential solutions are dominated by a previously obtained feasible solution. The bounding function tells the maximum overload of each soft resource in the previous feasible schedule. For a more complete description of the tree search routine see Ferreira and Hodgson [7].

2.2 Heuristics

There are two extreme ways of evaluating the tree to find a feasible schedule. On one hand there is implicit enumeration of the entire tree, i.e., a search without any interaction from the scheduler. Even for quite limited scheduling problems, the tree may be big and therefore result in excessively long computations. For large problems this approach may not be computationally feasible. Therefore, to implicitly enumerate the whole tree without intervention is probably not a good approach. On the other hand, the scheduler completely controls the search for a feasible solution and each job is assigned separately. The computer is used for evaluation of each assignment and for graphical output. This approach is time-consuming and may not be efficient either.

Neither of the two extreme approaches take full advantage of both man's and the computer's capabilities. A better utilization of man-machine interaction should be possible. In this study a number of heuristics have been used to improve the scheduling procedure. They can be divided into two types. The first one comprises heuristics that aim at limiting the size of the tree, the second one contains heuristics that make the search more efficient in terms of computation time.

2.2.1 Tree construction

In many planning situations, it might be desired to start a project at a specific time or to undertake a project during a certain time period, i.e., some start times for the project are more preferred than others. A specification of a desired start time can be used to immensely reduce the size of the tree. The length of the time period decides the number of branches at the corresponding level/node in the tree. By specifying for each project a desired start time and early and late start times, the number of branches at node i will be defined. These data may be based on project requirements or intuitively estimated from past experience.

2.2.2 Tree search

The efficiency in finding a solution to the scheduling problem is not only affected by how much the tree is reduced, but also by how the search in the tree is undertaken. It is more efficient to search through a tree when the top is as narrow as possible compared to the base. This can be achieved by sequencing the projects in ascending order according to size, i.e., the number of branches (the tree width). The role of the tree width may be interpreted as a priority assigned to each project. A high priority project may be given a small tree width and a low priority a large tree width. Looked upon this way the choice of tree width will be important in meeting different projects requirements.

Given a number of projects and their requests a feasible schedule may not always exist. However, it may be possible to violate some of the resource constraints. In this situation it may be advantageous to find a schedule that results in overloading of some of the resources at certain time periods. The question is, how close is it possible to come to a feasible schedule, i.e., it is desired to minimize the overloading of the soft constraints. More specifically, it is desired to minimize the maximum overload of the resources. This can be achieved by generating a number of schedules that are feasible with respect to the hard constraints. Each generated schedule will be better than previously generated ones, if such a schedule exists.

To avoid excessively long search times the scheduler sets a time limit to the search. After overflow of the time limit the computer stops and the scheduler decides whether to continue to search or not. Given the state of the search the scheduler may want to change data (e.g., trimming the tree, alter project data, modifying resource constraints) before a new search. To aid him in this work a series of graphs may be called (e.g., partial project schedule, completion of partial schedules, project table). These graphs are very helpful in guiding the scheduler on where changes in data may be most efficient for finding a better schedule.

2.3 Solution Procedure

The solution procedure is based on evaluating the tree, as specified above, using a tree search algorithm. A flowchart is given below (Figure 2.2). The notation used in the graph is the one employed by the computer program, which is written in BASIC. In Appendix II, the meaning of each symbol is stated. With this the graph, hopefully, is self-explanatory.

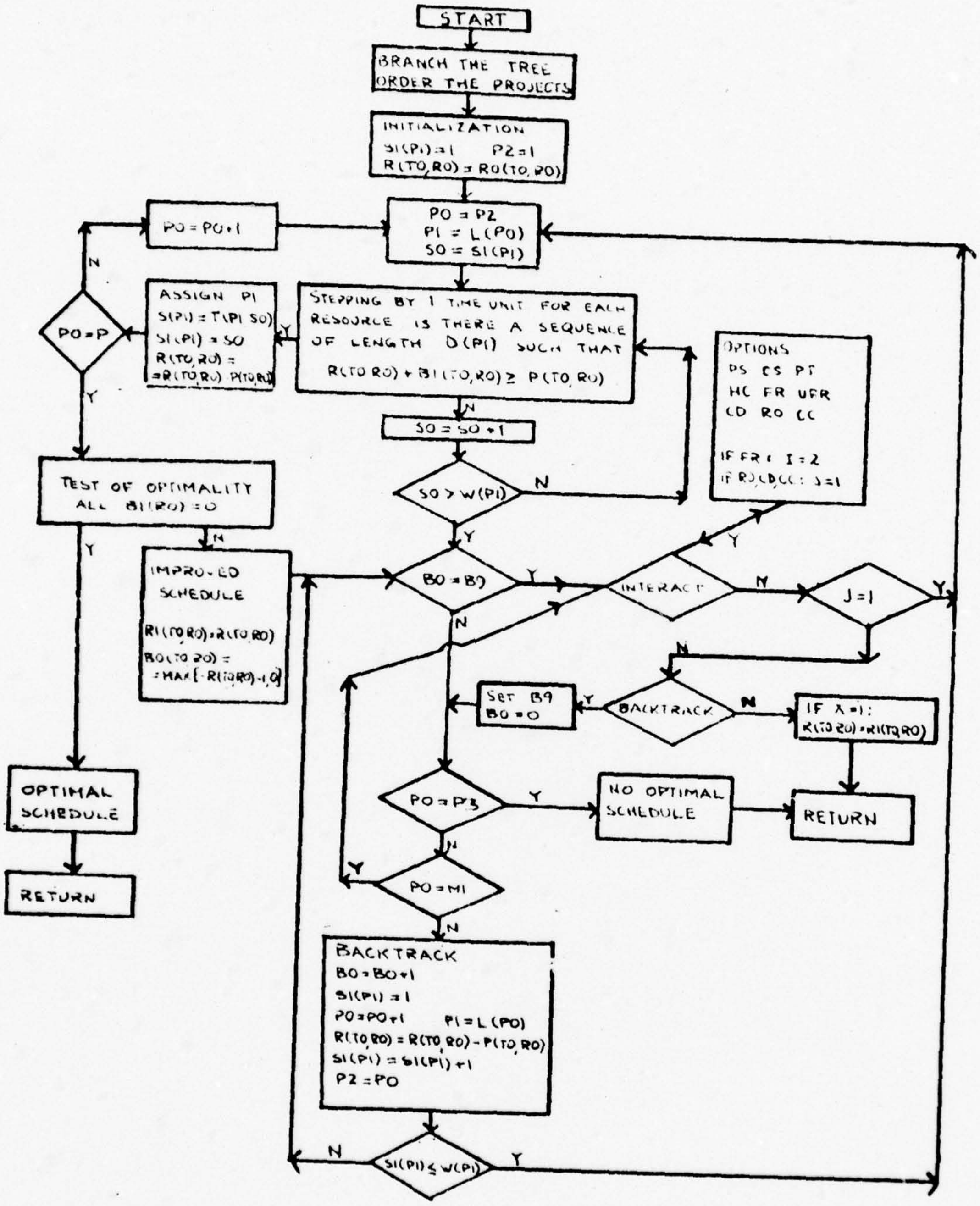


Figure 2.2. Flowchart of search algorithm

CHAPTER III

SYSTEM DESCRIPTION WITH EXAMPLES OF USE

3.1 Description of Computerized System

The basic hardware components of the system are a minicomputer (PDP 11/20), a video terminal (ADDS Consul 580) and a keyboard terminal (TI Silent 700 ASR). The program and input data are stored on a disk (Cal-data). The system is controlled via the minicomputer, the interaction between the computer and the scheduler takes place via the video terminal. The keyboard terminal is used for printing hard copies of selected output. The system configuration is shown in Figure 3.1.

The scheduling program, SCHED, is written in BASIC/RT 11. In Figure 3.2 a simplified picture is given of the program structure. The names in the boxes refer to the program elements in the BASIC code. Table 3.1 explains the functions of the different program elements.

3.2 Operation of System

The operation of the system starts with the loading of the program SCHED. The operator specifies what input data file he wants to use, and whether he wants to create or update a schedule. Then, the operator has to decide what

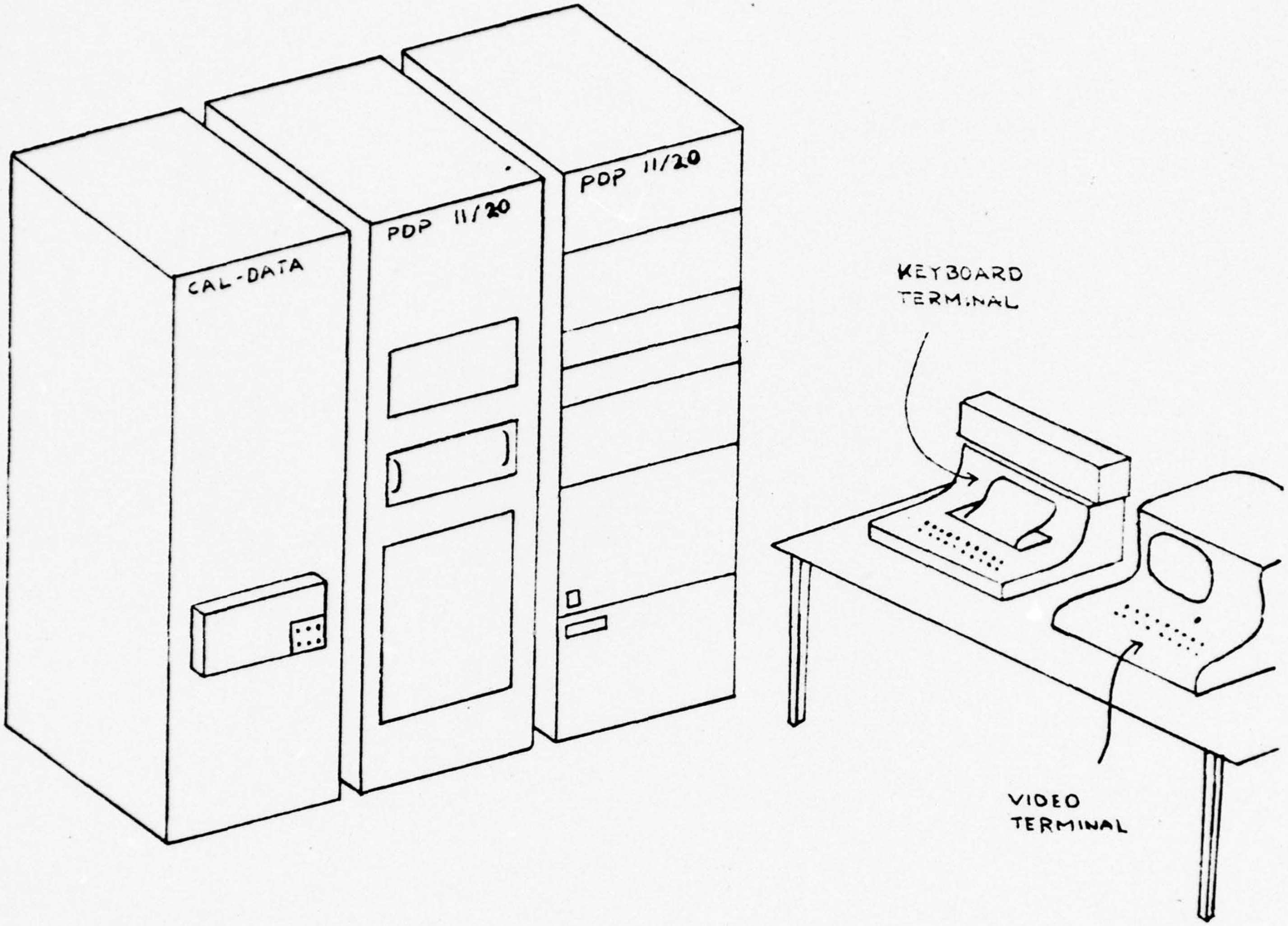


Figure 3.1 System Configuration

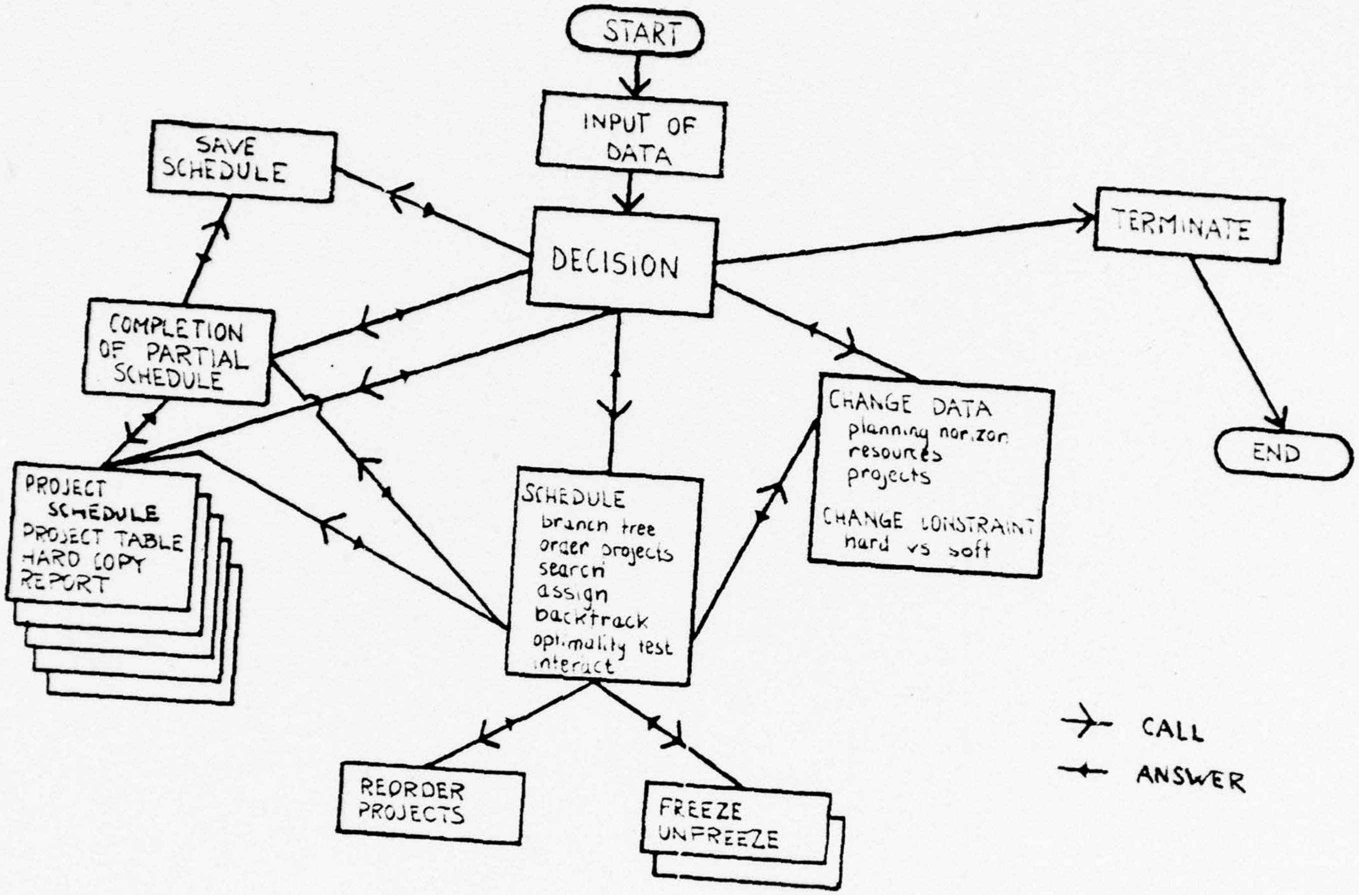

START
INPUT OF DATA
DECISION
SAVE SCHEDULE
COMPLETION OF PARTIAL SCHEDULE
TERMINATE
END
PROJECT SCHEDULE PROJECT TABLE HARD COPY REPORT
SCHEDULE branch tree order projects search assign backtrack optimality test interact
CHANGE DATA planning horizon resources projects
CHANGE CONSTRAINT hard vs soft
REORDER PROJECTS
FREEZE UNFREEZE
CALL
ANSWER

Figure 3.2 Program Structure

Table 3.1 Program Elements

Program Elements	Function
INPUT OF DATA	Reads the specified input file and initializes updating
DECISION	Controls the scheduling process
REORDER PROJECTS	Changes the order of the projects
CHANGE DATA	Changes planning horizon, resource and project data, as well as adds and cancels projects
CHANGE CONSTRAINT	Changes specified hard constraint into a soft one
COMPLETION OF PARTIAL SCHEDULE	Completes a partial schedule by assigning unassigned projects at desired start time
SCHEDULE	Forms the tree, searches through it for a feasible schedule, backtracks in the tree, minimizes overload of soft constraints, controls the search
PROJECT SCHEDULE	Forms and prints on the video screen a project schedule
PROJECT TABLE	Forms and prints on the video screen a project table
HARD COPY	Forms and prints a hard copy of selected output
SAVE SCHEDULE	Saves the schedule on specified file
REPORT	Forms and prints a standardized hard copy report
FREEZE	Stores partial schedule down to the current level of search
UNFREEZE	Retrieves the frozen partial schedule
TERMINATE	Terminates the scheduling process

he wants to do, e.g., he might start off with a look at a schedule based only on desired start times. This is done by calling the subroutine "Completion of partial schedule." By inspecting this he may get some hints on whether or not he should change some data or soften some constraints. If so he can call the subroutine "Change data" or "Change constraint." In some situations there may be a need to change data for other reasons, e.g., some data may have become out-of-date since the input file was created.

In most cases, the operator will probably not bother about changing data or constraints from the start. He wants to see if there is an easy available schedule before he is prepared to change the conditions. By calling the subroutine "Schedule" the search tree is formed and the search begun. If backtracking is needed to continue the search, the computer stops. The scheduler can now decide if he wants to continue the search by backtracking through the tree. Before he makes this decision, he has an opportunity to review the partial schedules at hand, and also complete these partial schedules. If he thinks it is worthwhile to continue the search without changing conditions, he specifies how long he wants the search to continue before next stop (since no clock is available a count of the number of backtracks is used). If the scheduler wants to change some conditions he calls the appropriate subroutines. After the changes have been made, the search is continued. When a feasible schedule is found control is automatically returned to "Decision."

In the case where there are soft constraints, the search routine has the objective of finding a schedule which minimizes the maximum resource overload. As in the case without soft constraints, the scheduler sets a time limit for the search. In addition he can review the best schedule found at the end of the time limit. If this is good enough, the search can stop, otherwise a new time limit is set. If the search continues long enough either a schedule with no overload will be found or the whole tree will be searched.

When the scheduler observes that he cannot find a good enough schedule within a reasonable amount of time, he has the opportunity to change both data and constraints. To guide him in this choice enlightened information may be found in the comparison of partial schedules with completed schedules, and by studying the available graphic displays (project schedule, project table). Some of the more powerful changes to be done may be to trim the tree and/or soften some resource constraints. It should be noted that these changes may be made within the search routine, but that only on the part of the tree "below" the current level of search.

As soon as the operator finds a satisfactory schedule, he may save this schedule on a file for future use or updating. By calling the subroutine "Save schedule" this can be achieved. The scheduler can also receive a hard copy of the schedule in a standardized report format.

What has been discussed so far concerns the main features in the operation of the system. In the subsequent section some more light will be shed on the interaction between man and computer. In the final section in this chapter, some examples of use will be given, where the various graphics will be shown.

3.3 Man-Computer Interaction

The main thrust of this study is, as stated earlier, to let the scheduler guide the computer in the search for a good schedule. By this guidance, the search will hopefully be more efficient and give a faster answer to the problem at hand. There is obviously some tradeoff between the amount of human interaction (interruption) and computer usage. If there is no human interaction, the computer usage may be very inefficient and total scheduling cost high. If the human interaction is considerable, the time and effort to find a satisfactory schedule also may be high. With an appropriate degree of human interaction the total scheduling cost may be lower and the quality of the result higher than otherwise possible.

With this in mind, the interactive elements in the program have been designed to give the scheduler as much freedom as possible, while attempting to maintain a high degree of computational efficiency. The interactive elements in the program allow the scheduler to

(1) completely change all input data about resource availability and project requirements

(2) add and cancel arbitrary projects

(3) give the search tree any form he likes, by changing desired start time, tree width and the ordering of the projects

(4) alter the constraint set by softening of specified constraints

(5) set a time limit for the tree search and take appropriate actions at these interrupts

(6) store and retrieve partial schedules down to the current level of search

(7) review actual partial schedules

(8) complete partial schedules

(9) get hard copies of selected graphics

(10) save schedules in a file for future use or update

Elements 3, 4, 5, and 6 may be considered as the main interactive elements, since they are very powerful, and the skill with which they are used will greatly influence the efficiency of the scheduling procedure. Elements 1, 2, and 10 are of special importance for updating of schedules, while elements 7, 8, and 9 are tools to help the scheduler in his analysis of what to do.

3.4 Examples of Use

3.4.1 Creating an initial schedule

Suppose the following scheduling problem is given. There are 9 projects that request to use the two available resources during the next 5 week planning period. The time unit to be used is working days (i.e., in this case the planning horizon is 25 days). Necessary data about resource availability and project requirements are stored in a file, PLA (Figure 3.3). When this file is created (using the program CREFIL), the scheduler specifies a desired start time and an early and late start time for each project. These data are arrived at from considerations based on project needs and priorities, preliminary estimates, past experience and tree search strategy. The more realistic the start times are and, consequently, the more limited the tree is, the faster the tree can be searched. It is therefore important not to assign these data totally arbitrarily. Some considerations should be taken into account. It is, however, probably not worthwhile to put too much effort into these considerations.

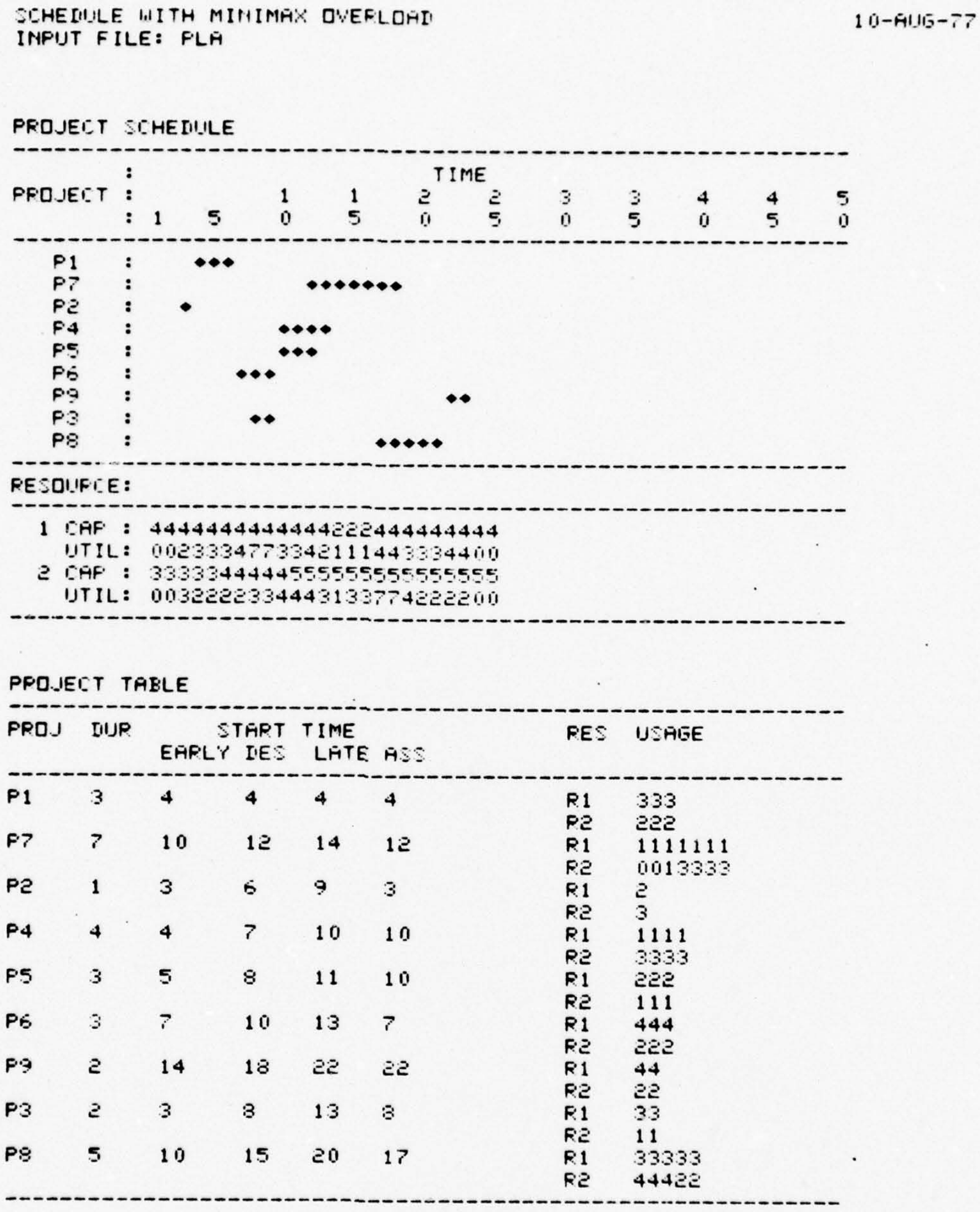
The available amount and the requested amount of resources may change over time in any pattern. In the problem at hand, there is maintenance work planned during days 14 to 16 for resource R1. Only two units will be available during this period. Resource R2 will not be fully available during the first two weeks due to repairs. Projects P7 and P8 will use a varying amount of resource R2.

```
SCHEDULE WITH DESIRED START TIMES                           10-AUG-77
INPUT FILE: PLA


PROJECT SCHEDULE
------------------------------------------------------------
        :                     TIME
PROJECT :          1    1    2    2    3    3    4    4    5
        : 1   5    0    5    0    5    0    5    0    5    0
------------------------------------------------------------
   P1   :    ◆◆◆
   P2   :      ◆
   P3   :        ◆◆
   P4   :       ◆◆◆◆
   P5   :        ◆◆◆
   P6   :          ◆◆◆
   P7   :            ◆◆◆◆◆◆◆
   P8   :               ◆◆◆◆◆
   P9   :                  ◆◆
------------------------------------------------------------
RESOURCE:
------------------------------------------------------------
  1 CAP : 4444444444444222444444444
    UTIL: 0003351667451144487000000
  2 CAP : 3333444444555555555555555
    UTIL: 0002253556220177774000000
------------------------------------------------------------


PROJECT TABLE
------------------------------------------------------------
PROJ  DUR      START TIME               RES  USAGE
           EARLY DES  LATE ASS
------------------------------------------------------------
P1    3    4     4    4    4            R1   333
                                        R2   222
P2    1    3     6    9    6            R1   2
                                        R2   3
P3    2    3     8    13   8            R1   33
                                        R2   11
P4    4    4     7    10   7            R1   1111
                                        R2   3333
P5    3    5     8    11   8            R1   222
                                        R2   111
P6    3    7     10   13   10           R1   444
                                        R2   222
P7    7    10    12   14   12           R1   1111111
                                        R2   0013333
P8    5    10    15   20   15           R1   33333
                                        R2   44422
P9    2    14    18   22   18           R1   44
                                        R2   22
------------------------------------------------------------
```

Figure 3.3 Schedule with desired start times

The input data file was read into the program from file PLA. All data are up-to-date and no soft-constraints are considered yet, so subroutine "Schedule" is called. Backtracking is needed. After searching through the whole tree, it is found that no feasible schedule exists. Obviously, the search could have been stopped before the whole tree had been searched. In fact, it is probably more likely to stop as soon as it is realized that a feasible schedule might not exist or will not be found without a long search. In this example, the tree is very limited and it did not take any unreasonably long time to search the tree, so for demonstration purposes, the search was continued until all possibilities were scrutinized.

To find a feasible schedule, there are two choices. First, some of the constraints may be softened and a search undertaken that minimizes the maximum overloading of the resources. By inspection of the completed schedule based on desired start times (Figure 3.3), it is found that it is very likely that the most severe conflicts concerning resource R1 will be during time period 8 to 10, and concerning resource R2 during time period 15 to 18. Other resource conflicts are also at hand, e.g., resource 1 during time period 18 to 19. One might expect, however, that this conflict could be resolved by a shift of some start times. The maximum overload is 4 units. Using the minimax search procedure, the schedule in Figure 3.4 was found, where the maximum overload is 3 units. This may not be satisfactory, so one might try a second choice.

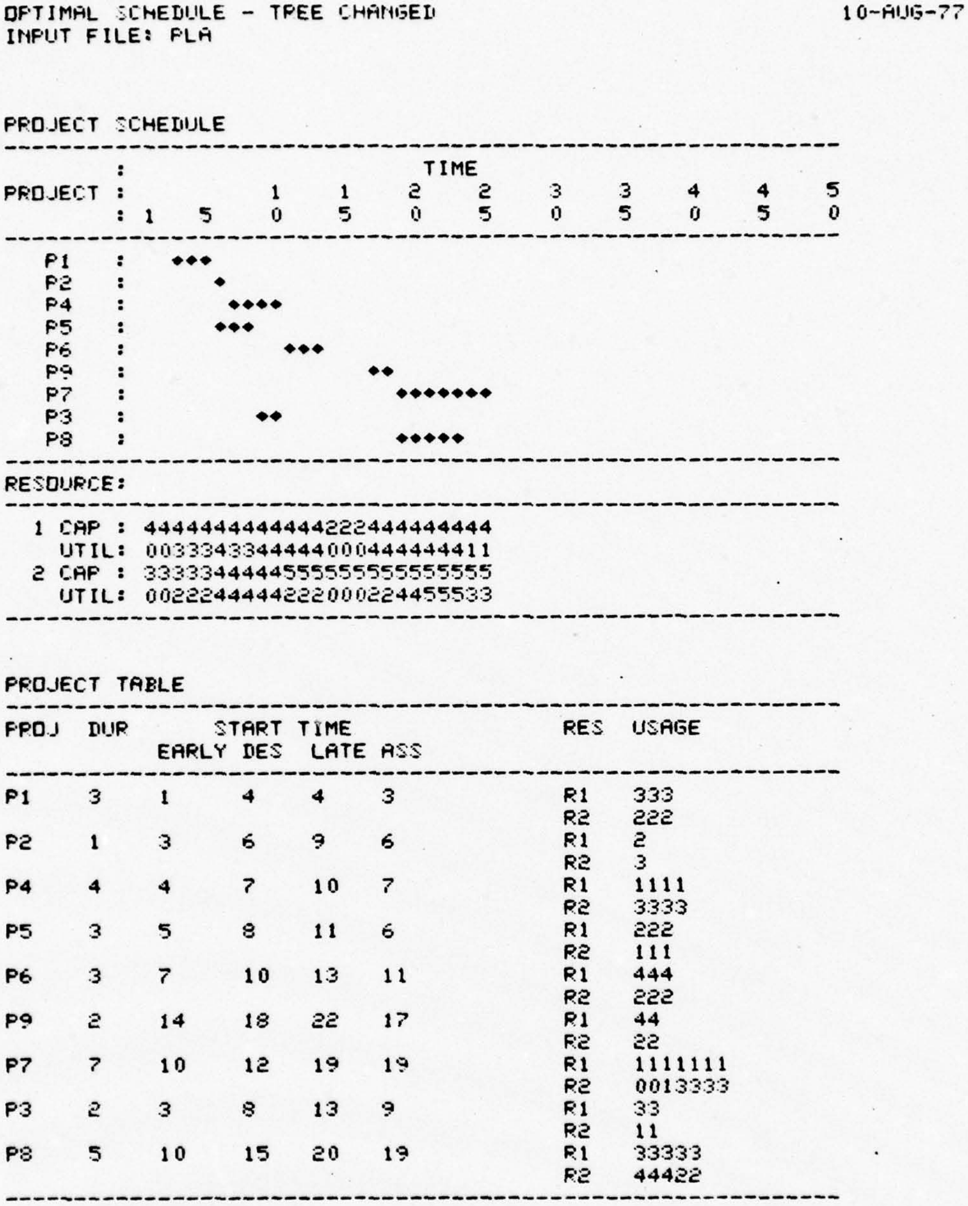


```
SCHEDULE WITH MINIMAX OVERLOAD                                    10-AUG-77
INPUT FILE: PLA


PROJECT SCHEDULE
-------------------------------------------------------------
        :                    TIME
PROJECT :         1    1    2    2    3    3    4    4    5
        : 1   5    0    5    0    5    0    5    0    5    0
-------------------------------------------------------------
   P1   :    ◆◆◆
   P7   :            ◆◆◆◆◆◆◆
   P2   :   ◆
   P4   :          ◆◆◆◆
   P5   :          ◆◆◆
   P6   :       ◆◆◆
   P9   :                      ◆◆
   P3   :        ◆◆
   P8   :                 ◆◆◆◆◆
-------------------------------------------------------------
RESOURCE:
-------------------------------------------------------------
  1 CAP : 4444444444444222444444444
    UTIL: 0023334773342111443334400
  2 CAP : 3333344444555555555555555
    UTIL: 0032222334443133774222200
-------------------------------------------------------------


PROJECT TABLE
-------------------------------------------------------------
PROJ  DUR        START TIME                     RES   USAGE
           EARLY DES   LATE ASS
-------------------------------------------------------------
P1     3    4     4     4    4                  R1    333
                                                R2    222
P7     7    10    12    14   12                 R1    1111111
                                                R2    0013333
P2     1    3     6     9    3                  R1    2
                                                R2    3
P4     4    4     7     10   10                 R1    1111
                                                R2    3333
P5     3    5     8     11   10                 R1    222
                                                R2    111
P6     3    7     10    13   7                  R1    444
                                                R2    222
P9     2    14    18    22   22                 R1    44
                                                R2    22
P3     2    3     8     13   8                  R1    33
                                                R2    11
P8     5    10    15    20   17                 R1    33333
                                                R2    44422
-------------------------------------------------------------
```

Figure 3.4 Schedule with minimax overload

It is observed from the project schedule that resource R1 is heavily used around time period 10, and that there is no utilization at the beginning of the planning period. By decreasing the early start time by 3 units for project P1 it may be possible to shift it to the left and ease the requests later on. For resource R2 there is a conflict during time period 17 to 18. By increasing the late start time for project P7 by 5 units it may be possible to shift P7 to the right. With these two changes of the tree, another search is undertaken. The result is an optimal schedule, i.e., a feasible schedule with no soft constraints, or maximum overload equal to zero. The schedule is described in Figure 3.5 in the standardized report form. The found schedule is now saved in a data file for convenient future updating purposes.

In the search for the optimal schedule above, all interaction in the scheduling process took place outside the search routine except for setting time limits for the search. The operator did not change the conditions for the search once it was started. He changed data and constraints before and after entering the search routine. This will be the most efficient degree of interaction in many cases. As the problem size increases and an optimal schedule is less easily found, it may be advantageous to interact within the search routine itself. This possibility will now be illustrated for the above problem. Since this is a rather simple problem it would not normally be justified to employ this interaction

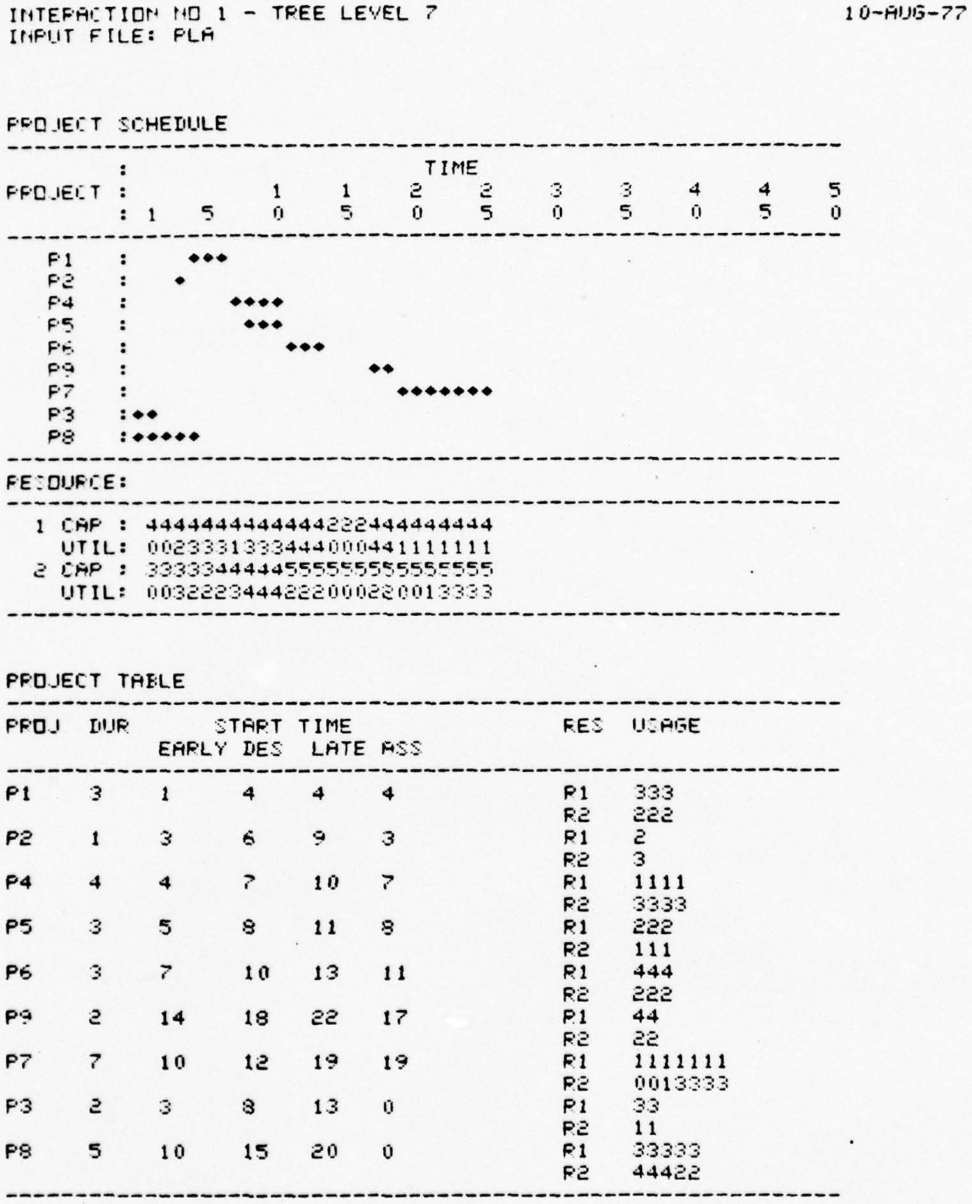


```
OPTIMAL SCHEDULE - TREE CHANGED                             10-AUG-77
INPUT FILE: PLA


PROJECT SCHEDULE
------------------------------------------------------------
        :                     TIME
PROJECT :          1    1    2    2    3    3    4    4    5
        : 1   5    0    5    0    5    0    5    0    5    0
------------------------------------------------------------
   P1   :   ◆◆◆
   P2   :      ◆
   P4   :       ◆◆◆◆
   P5   :      ◆◆◆
   P6   :           ◆◆◆
   P9   :                 ◆◆
   P7   :                   ◆◆◆◆◆◆◆
   P3   :         ◆◆
   P8   :                   ◆◆◆◆◆
------------------------------------------------------------
RESOURCE:
------------------------------------------------------------
  1 CAP : 4444444444444222444444444
    UTIL: 0033343344444000444444411
  2 CAP : 3333344444555555555555555
    UTIL: 0022244444222000224455533
------------------------------------------------------------


PROJECT TABLE
------------------------------------------------------------
PROJ  DUR      START TIME              RES  USAGE
           EARLY DES  LATE ASS
------------------------------------------------------------
P1    3    1     4    4    3            R1   333
                                        R2   222
P2    1    3     6    9    6            R1   2
                                        R2   3
P4    4    4     7    10   7            R1   1111
                                        R2   3333
P5    3    5     8    11   6            R1   222
                                        R2   111
P6    3    7     10   13   11           R1   444
                                        R2   222
P9    2    14    18   22   17           R1   44
                                        R2   22
P7    7    10    12   19   19           R1   1111111
                                        R2   0013333
P3    2    3     8    13   9            R1   33
                                        R2   11
P8    5    10    15   20   19           R1   33333
                                        R2   44422
------------------------------------------------------------
```

Figure 3.5 Optimal schedule

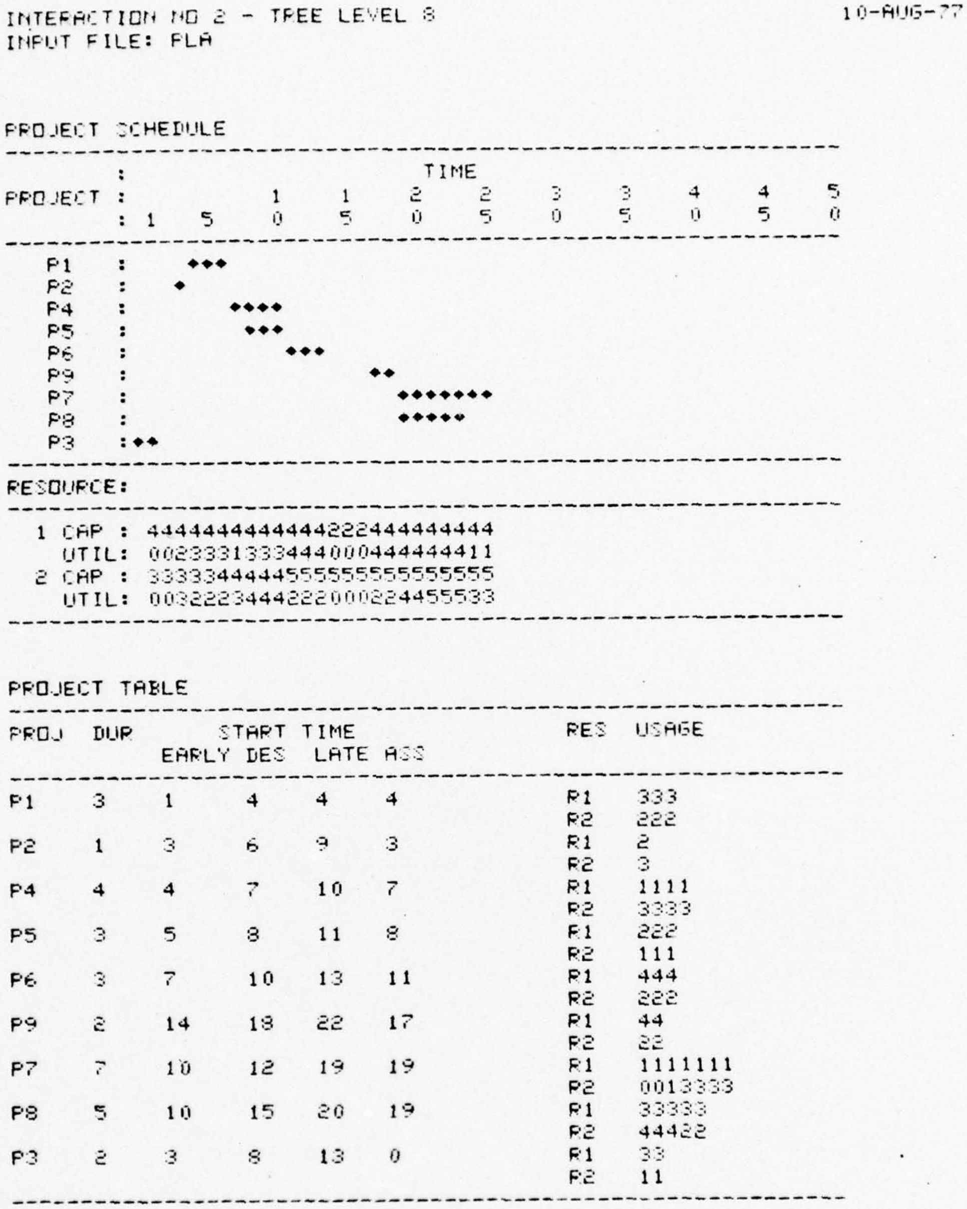
approach. The justification here is to illustrate what, in fact, can be done. How to properly undertake the interaction is ultimately up to the judgement of the operator and is outside the scope of this study.

As in the previous example, the schedule based on desired start times is inspected (Figure 3.3) and the start times for projects P1 and P7 changed. By calling the search subroutine, level 4 in the tree is reached before the computer stops (i.e., it is necessary to backtrack). Since the search stopped quite high up in the tree, it is decided to continue the tree search in order to get further down in the tree. However, it is desired to stop the search as soon as a lower tree level is reached. To achieve this, a tree level test is activated. Each time it is necessary to backtrack a check is performed to determine if the actual tree level is lower than the lowest tree level obtained previously. After continuing the search a while, level 7 is reached. Reviewing the schedule (Figure 3.6) we find that projects P3 and P8 are unassigned. The partial schedule is frozen down to level 7 (i.e., the tree search will be limited to the part of the tree below level 7). After interchanging the order of project P3 and P8 another search is undertaken. This time we will reach level 8 in the tree (Figure 3.7). However, it is not possible to fit in project P3 into the partially frozen tree. The search returns to the level where the tree was frozen and the state of the search is restored by unfreezing the tree. Obviously, changes must be made further up in the tree in



```
INTERACTION NO 1 - TREE LEVEL 7                             10-AUG-77
INPUT FILE: PLA


PROJECT SCHEDULE
------------------------------------------------------------
        :                     TIME
PROJECT :          1    1    2    2    3    3    4    4    5
        : 1   5    0    5    0    5    0    5    0    5    0
------------------------------------------------------------
   P1   :    ◆◆◆
   P2   :   ◆
   P4   :       ◆◆◆◆
   P5   :        ◆◆◆
   P6   :           ◆◆◆
   P9   :                 ◆◆
   P7   :                   ◆◆◆◆◆◆◆
   P3   :◆◆
   P8   :◆◆◆◆◆
------------------------------------------------------------
RESOURCE:
------------------------------------------------------------
  1 CAP : 4444444444444222444444444
    UTIL: 0023331333444000441111111
  2 CAP : 3333344444555555555555555
    UTIL: 0032223444222000220013333
------------------------------------------------------------


PROJECT TABLE
------------------------------------------------------------
PROJ  DUR      START TIME                RES  USAGE
           EARLY DES  LATE ASS
------------------------------------------------------------
P1    3    1     4    4    4            R1   333
                                        R2   222
P2    1    3     6    9    3            R1   2
                                        R2   3
P4    4    4     7    10   7            R1   1111
                                        R2   3333
P5    3    5     8    11   8            R1   222
                                        R2   111
P6    3    7     10   13   11           R1   444
                                        R2   222
P9    2    14    18   22   17           R1   44
                                        R2   22
P7    7    10    12   19   19           R1   1111111
                                        R2   0013333
P3    2    3     8    13   0            R1   33
                                        R2   11
P8    5    10    15   20   0            R1   33333
                                        R2   44422
------------------------------------------------------------
```

Figure 3.6 Interaction no. 1



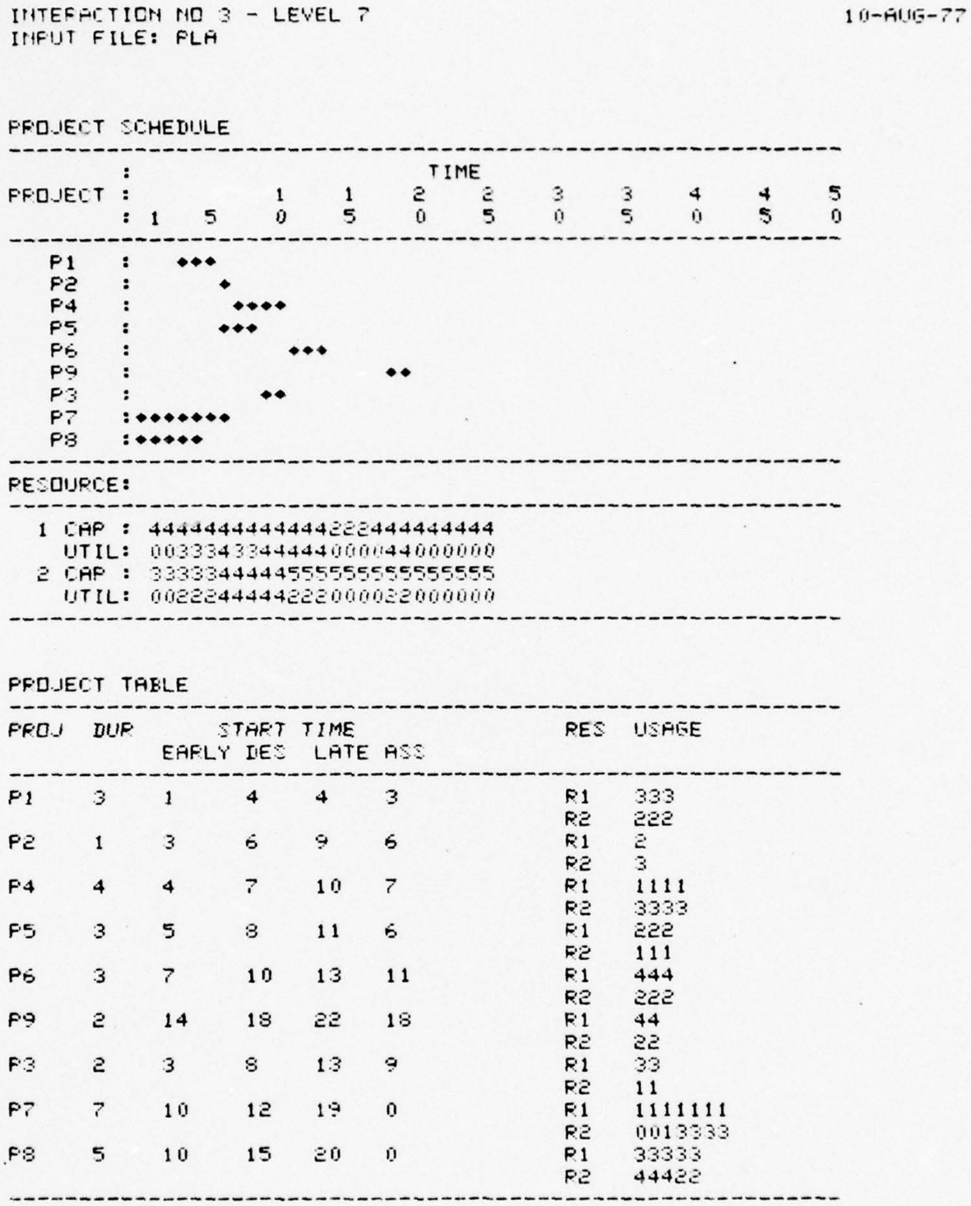
```
INTERACTION NO 2 - TREE LEVEL 8                                      10-AUG-77
INPUT FILE: PLA


PROJECT SCHEDULE
------------------------------------------------------------------------------
          :                          TIME
PROJECT   :              1     1     2     2     3     3     4     4     5
          : 1   5        0     5     0     5     0     5     0     5     0
------------------------------------------------------------------------------
   P1     :    ◆◆◆
   P2     :   ◆
   P4     :       ◆◆◆◆
   P5     :        ◆◆◆
   P6     :           ◆◆◆
   P9     :                ◆◆
   P7     :                  ◆◆◆◆◆◆◆
   P8     :                  ◆◆◆◆◆
   P3     :◆◆
------------------------------------------------------------------------------
RESOURCE:
------------------------------------------------------------------------------
  1 CAP : 4444444444444222444444444
    UTIL: 0023331333444000444444411
  2 CAP : 3333444444555555555555555
    UTIL: 0032223444222000224455533
------------------------------------------------------------------------------


PROJECT TABLE
------------------------------------------------------------------------------
PROJ  DUR        START TIME                           RES  USAGE
            EARLY DES   LATE ASS
------------------------------------------------------------------------------
P1     3     1     4     4     4                      R1    333
                                                      R2    222
P2     1     3     6     9     3                      R1    2
                                                      R2    3
P4     4     4     7     10    7                      R1    1111
                                                      R2    3333
P5     3     5     8     11    8                      R1    222
                                                      R2    111
P6     3     7     10    13    11                     R1    444
                                                      R2    222
P9     2     14    18    22    17                     R1    44
                                                      R2    22
P7     7     10    12    19    19                     R1    1111111
                                                      R2    0013333
P8     5     10    15    20    19                     R1    33333
                                                      R2    44422
P3     2     3     8     13    0                      R1    33
                                                      R2    11
------------------------------------------------------------------------------
```

Figure 3.7 Interaction no. 2

order to find an optimal schedule (which we know exists from the previous example). Project P3 is moved up in the tree and fit in immediately before project P7. After continuing the search a while, level 7 is reached. By inspecting the schedule (Figure 3.8) it is noticed that by changing the start time for project P9 from time period 18 to 17 it might be possible to fit in project P7. After performing this change the search is continued and an optimal schedule is found without further backtracking (Figure 3.5).

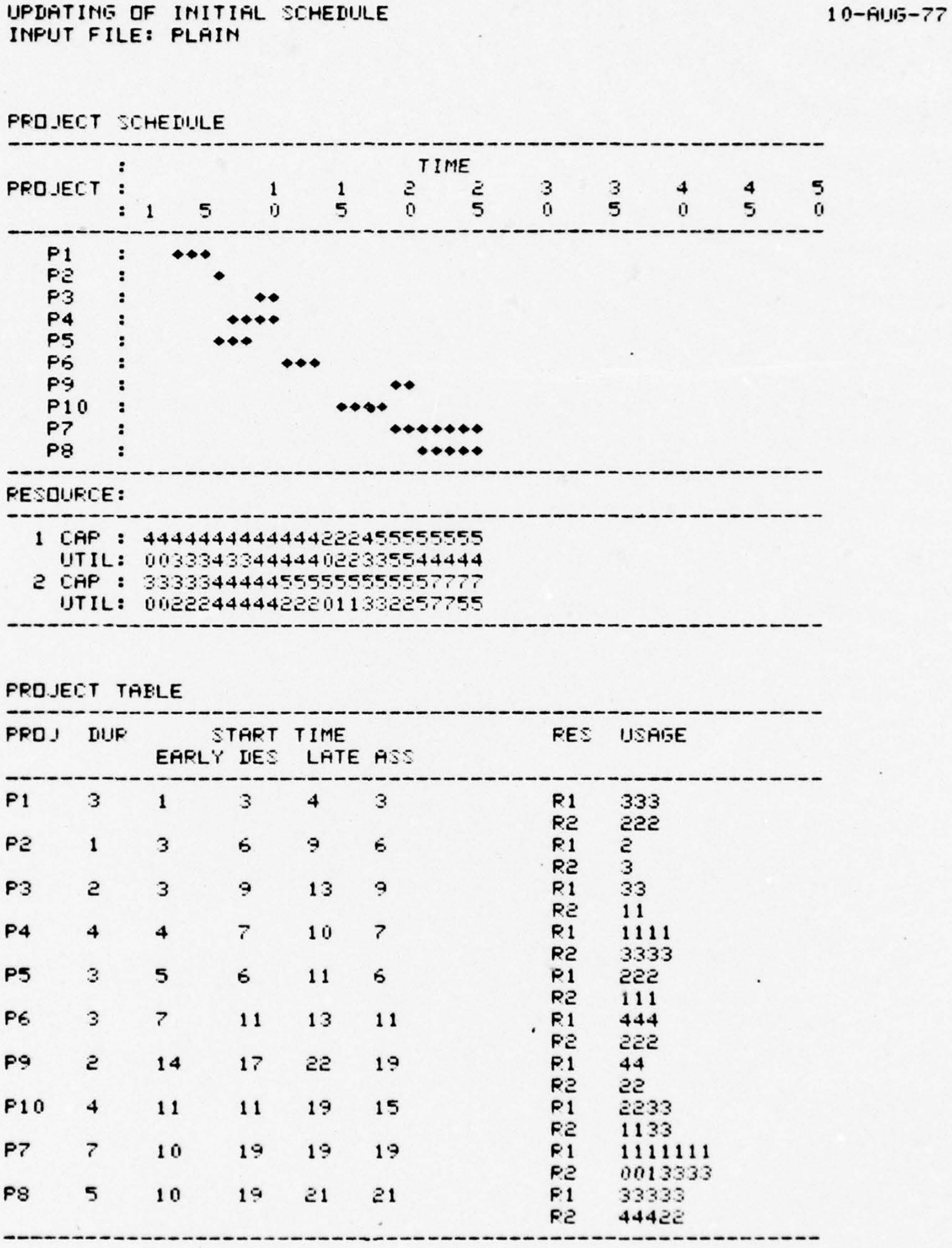
3.4.2 Updating and initial schedule

At the end of the second week of the initial schedule above, a request is made by a new project, P10. It is necessary to start this project between time period 11 and 19. An early start time is preferred. This implies a desired start time equal to 11 and a late start time of 19. Since the initial schedule was created, it has also become certain that more units will be added to the resources at the end of the planning horizon. From time period 18 and on there will be one more unit available of resource R1. From time period 22 and on resource R2 will have two more units available. After taking these changes into consideration, a new search is undertaken from time period 11. An optimal schedule is found, but at the expense of shifting the start of project P8 from time period 19 to 21 and P9 from time period 17 to 19 (Figure 3.9). Observe that projects with a start time prior to time period 11 have not been moved. The up-



```
INTERACTION NO 3 - LEVEL 7                                  10-AUG-77
INPUT FILE: PLA


PROJECT SCHEDULE
------------------------------------------------------------
        :                     TIME
PROJECT :          1    1    2    2    3    3    4    4    5
        : 1   5    0    5    0    5    0    5    0    5    0
------------------------------------------------------------
   P1   :   ◆◆◆
   P2   :      ◆
   P4   :       ◆◆◆◆
   P5   :      ◆◆◆
   P6   :           ◆◆◆
   P9   :                  ◆◆
   P3   :         ◆◆
   P7   :◆◆◆◆◆◆◆
   P8   :◆◆◆◆◆
------------------------------------------------------------
RESOURCE:
------------------------------------------------------------
  1 CAP : 4444444444444222444444444
    UTIL: 0033343344444000044000000
  2 CAP : 3333344444555555555555555
    UTIL: 0022244444222000022000000
------------------------------------------------------------


PROJECT TABLE
------------------------------------------------------------
PROJ  DUR       START TIME                      RES  USAGE
           EARLY DES  LATE ASS
------------------------------------------------------------
P1     3    1     4    4    3                   R1   333
                                                R2   222
P2     1    3     6    9    6                   R1   2
                                                R2   3
P4     4    4     7    10   7                   R1   1111
                                                R2   3333
P5     3    5     8    11   6                   R1   222
                                                R2   111
P6     3    7     10   13   11                  R1   444
                                                R2   222
P9     2    14    18   22   18                  R1   44
                                                R2   22
P3     2    3     8    13   9                   R1   33
                                                R2   11
P7     7    10    12   19   0                   R1   1111111
                                                R2   0013333
P8     5    10    15   20   0                   R1   33333
                                                R2   44422
------------------------------------------------------------
```

Figure 3.8 Interaction no. 3



```
UPDATING OF INITIAL SCHEDULE                                     10-AUG-77
INPUT FILE: PLAIN


PROJECT SCHEDULE
------------------------------------------------------------
        :                      TIME
PROJECT :          1    1    2    2    3    3    4    4    5
        : 1   5    0    5    0    5    0    5    0    5    0
------------------------------------------------------------
   P1   :   ◆◆◆
   P2   :      ◆
   P3   :         ◆◆
   P4   :       ◆◆◆◆
   P5   :      ◆◆◆
   P6   :           ◆◆◆
   P9   :                   ◆◆
   P10  :               ◆◆◆◆
   P7   :                   ◆◆◆◆◆◆◆
   P8   :                     ◆◆◆◆◆
------------------------------------------------------------
RESOURCE:
------------------------------------------------------------
  1 CAP : 4444444444444222455555555
    UTIL: 0033343344444022335544444
  2 CAP : 3333344444555555555557777
    UTIL: 0022244444222011332257755
------------------------------------------------------------


PROJECT TABLE
------------------------------------------------------------
PROJ  DUP        START TIME                     RES   USAGE
           EARLY DES  LATE ASS
------------------------------------------------------------
P1     3    1     3    4    3                   R1    333
                                                R2    222
P2     1    3     6    9    6                   R1    2
                                                R2    3
P3     2    3     9    13   9                   R1    33
                                                R2    11
P4     4    4     7    10   7                   R1    1111
                                                R2    3333
P5     3    5     6    11   6                   R1    222
                                                R2    111
P6     3    7     11   13   11                  R1    444
                                                R2    222
P9     2    14    17   22   19                  R1    44
                                                R2    22
P10    4    11    11   19   15                  R1    2233
                                                R2    1133
P7     7    10    19   19   19                  R1    1111111
                                                R2    0013333
P8     5    10    19   21   21                  R1    33333
                                                R2    44422
------------------------------------------------------------
```

Figure 3.9 Updating of initial schedule

dated schedule is saved in a file for future use or renewed updating.

CHAPTER IV

CONCLUSIONS

The system developed above approaches the solution to the given scheduling problem through the extensive use of man-computer interaction. The operator interacts with the computer to guide in the search for an optimal schedule. This guidance takes the form of changing conditions and of directing the search through an implicit enumeration tree. The basis for this guidance is obtained by the operator through the viewing and intuitive analysis of the different graphics and tables provided by the system. The system is very powerful in that it gives the operator great possibilities to control the scheduling process.

The scheduling model itself has one abstraction. It is assumed that all units of a certain type of resource are interchangeable. For example, if there are three units of resource R1 available in one time period and exactly the same number in the following period, these units may not be the same even though they are of the same kind. In practical situations, it might be impossible to interchange the units. Suppose that 1 unit of resource R1 is requested by project P1 during five days. For the first three days there are several units of resource R1 available, but they have to be sent for maintenance work at the beginning of the fourth day. At the end of the third day, however, several units of resource R1 will return from maintenance work. Thus, though there are enough units of resource R1 available, due to noninterchangeability it might be impossible to satisfy the request. How to handle this situation efficiently has to be studied further.

The system developed has the potential to be useful in real life applications. The tool is developed. Now it is necessary to refine the tool and experiment with it to find the best ways of using it in different situations. There is a good deal of flexibility built into the system. It is necessary to determine how this flexibility can be used when dealing with problems of different size and complexity. What kind of strategy is appropriate for operator interaction? When is it most advisable to interact and what type of interactions is most appropriate? There are probably no definite answers to questions of this type, but rules for using the system in an efficient way need to be developed. Turn around time and scheduling cost considerations will play an important role in this development.

The results so far achieved are promising for real world applications. The reason for this lies in the basic approach taken in building the system. In many real scheduling situations, there is a great deal of uncertainty as to what will happen during future planning periods. Plans made have to be changed as conditions change. Alternate plans have to be developed. The system as developed, aims at handling such situations and at producing acceptable schedules. Theoretically exact solutions are not of primary interest. The uncertainty of future events requires flexibility in generating schedules. It is important to find good approximate solutions to the scheduling problem and to study alternate schedules. The system provides a tool to operate in that environment.

BIBLIOGRAPHY

1. Anderson, D. C., Stowell, G. W., and Langhorst, F. E., "Computer Graphics," Detroit Engineer, Oct. 1976.

2. Balas, E., "Project Scheduling with Resource Constraints," chapter in Applications of Mathematical Programming, Beale, E. M. L., English University Press, London, 1971.

3. Conguet, L., "Comparison of Three Heuristic Procedures for Allocating Resources and Producing Schedules," from Lombaers, H. J. M., "Project Planning by Network Analysis," Proc. Second Int. Cong. of Internet, North-Holland, Amsterdam, 1969.

4. Conway, R. W., Maxwell, W. L., and Miller, L. W., Theory of Scheduling, Addison Wesley, Reading, Mass., 1967.

5. Davies, E. W., "Project Scheduling under Resource Constraints - Historical Review and Categorization of Procedures," AIIE Transactions, 1967.

6. Davies, E. W. and Heidorn, G. E., "An Algorithm for Optimal Project Scheduling under Multiple Resource Constraints," Management Science, Vol. 17, 1971.

7. Ferreira, A. C. and Hodgson, T. J., "An N-Product, Multi-Machine Lotsize Scheduling Model," AIIE Transactions, Vol. 5, No. 3, 1973.

8. Franke, G., "Zur Ablaufplanung in Netzwerken (Scheduling in Networks)," Z. fur Betriebswirtschaftliche Forschung, 23, 1971.

9. Herroelen, W. S, "Resource Constrained Project Scheduling - The State of the Art," Operational Research Quarterly, Vol. 23, No. 3, 1972.

10. Johnson, T. J. R., "A Branch and Bound Algorithm for the Resource Constrained Project Scheduling Problem," Research Report, M.I.T., 1967.

11. Leahy, J. P., "A Man-Machine CPM System for Decision Making in the Construction Industry," Construction Research Series No. 9, Dept. of Civil Engineering, University of Illinois, Urbana, 1967.

12. Mandeville, D. W., "The Development of Network Analysis Resource Balancing Methods from Assembly Line Balancing Techniques," unpublished MS thesis, Purdue University, Indiana, 1965.

13. Muth, J. F. and Thompson, G. L., Industrial Scheduling, Prentice-Hall, Englewood Cliffs, N.J., 1963, Ch. 21.

14. Paulson, B. C., Jr., "Man-Computer Concepts for Project Management," Technical Report No. 148, Dept. of Civil Engineering, Stanford University, Stanford, Calif., 1971.

15. Pouring, A. A. and Sladky, J. F., "Computers and Technology," Mechanical Engineering, May 1977.

16. Prabhakar, T., "Resource Allocation/Leveling in Engineering/Construction Projects," Paper presented at the 37th Meeting of ORSA, 20-22 April, 1970.

17. Pritsker, A. A. B., Watters, L. J., and Wolfe, P. M., "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach," Management Science, Vol. 16, 1969.

18. Pritsker, A. A. B. and Watters, L. J., "A Zero-One Programming Approach to Scheduling with Limited Resources," Rand Corp., Paper RM-5561-PR.

19. Schrage, L., "Solving Resource Constrained Network Problems by Implicit Enumeration - Nonpreemptive Case," Operations Research, No. 18, 1970.

20. Sisson, R. L., "Sequencing Theory," Ch. 7 of Progress in Operations Research, R. Ackoff, Ed., Wiley, New York, N.Y., 1969.

21. Wiest, J. D., "Toward a Man-Machine Interactive System for Project Scheduling," Lecture Notes in Economics and Mathematical System, No. 86 ("Symposium on the Theory of Scheduling and Its Applications"), Springer-Verlag, New York, N.Y., 1973.

22. Wiest, J. D., "A Heuristic Model for Scheduling Large Projects with Limited Resources," Management Science, Vol. 13, No. 6, Feb. 1967.

23. Wilson, R. C., "Assembly Line Balancing and Resource Leveling," Production and Inventory Control, University of Michigan, Engineering Summer Conferences, 1964.

APPENDIX I

SOURCE LISTING

The computer program is written in BASIC/RT 11 (DEC-11-LBACA-D-D). In Appendix II the variables used are listed and explained. In Appendix III, the functions of the different program elements are described.

```
SCHEDH.BAS           10-AUG-77                                     PAGE 1

20 REM        ************************************
40 REM        *  THIS IS AN INTERACTIVE HEURISTIC  *
60 REM        * PROCEDURE FOR RESOURCE CONSTRAINED *
80 REM        *       SCHEDULING OF PROJECTS       *
100 REM       ************************************
120 REM       *           PER-OLOF CARLSON          *
140 REM       *             AUGUST 1977             *
160 REM       ************************************
180 REM
200 REM
220 DIM R(40,3),R0(40,3),R1(40,3)
240 DIM P$(10),D(10),S(10),S0(10),S1(10)
260 DIM P(100,3)
280 DIM T(10,25)
300 DIM W(10),W0(10),W1(10)
320 DIM L(10)
340 DIM B0(40,3),B1(40,3)
360 REM *****************
380 REM * INPUT OF DATA *
400 REM *****************
420 PRINT CHR$(1)\PRINT CHR$(12)
440 PRINT "NAME OF INPUT FILE";\INPUT F$
520 OPEN F$ FOR INPUT AS FILE VF1$(1000)
540 REM
560 REM *** RESOURCE MATRIX ***
580 REM
600 T=VAL(VF1(1))
620 R=VAL(VF1(2))
640 P=VAL(VF1(3))
660 FOR R0=1 TO R
680 V0=R0*50
700 R$(R0)=VF1(V0)
720 FOR T0=1 TO T
740 R0(T0,R0)=VAL(VF1(V0+T0))
760 B0(T0,R0)=0
780 NEXT T0
800 NEXT R0
820 REM
840 REM *** PROJECT MATRIX ***
860 REM
880 D(0)=0\K(0)=0
900 FOR P0=1 TO P
920 V0=200+P0*50
940 P$(P0)=VF1(V0)
960 D(P0)=VAL(VF1(V0+1))
980 S0(P0)=VAL(VF1(V0+2))
1000 W0(P0)=VAL(VF1(V0+3))
1020 W1(P0)=VAL(VF1(V0+4))
1030 W(P0)=W0(P0)+W1(P0)+1
1040 V0=V0+5
1060 K(P0)=K(P0-1)+D(P0-1)
1080 FOR R0=1 TO R
1100 FOR D0=1 TO D(P0)
1120 V0=V0+1
1140 D1=D0+K(P0)
1160 P(D1,R0)=VAL(VF1(V0))
1180 NEXT D0
1200 NEXT R0
1220 NEXT P0
1230 CLOSE VF1
```

```
SCHEDH.BAS            10-AUG-77                          PAGE  2

1240 T5=1\P3=1
1260 PRINT "DO YOU WANT TO UPDATE A PREVIOUS SCHEDULE";\INPUT A$
1280 IF SEG$(A$,1,1)="N"GO TO 1560
1300 IF SEG$(A$,1,1)<>"Y"GO TO 1260
1320 REM
1340 REM ◆◆◆ UPDATE ◆◆◆
1360 REM
1380 PRINT "UPDATING STARTS AT TIME";\INPUT T5
1400 IF T5<=TGO TO 1440
1420 PRINT "WRONG TIME - TRY AGAIN"\GO TO 1380
1440 IF T5>1GO TO 1480 \RETURN
1480 FOR P0=1 TO P
1500 IF S0(P0)>=T5GO TO 1540
1520 W(P0)=-1\P3=P3+1
1540 NEXT P0
1560 X=0
1580 P2=1\F=0\Y=0
1600 FOR R0=1 TO R\FOR T0=1 TO T
1620 R(T0,R0)=R0(T0,R0)\NEXT T0\NEXT R0
1640 REM ◆◆◆◆◆◆◆◆◆◆◆◆◆
1660 REM ◆ DECISION ◆
1680 REM ◆◆◆◆◆◆◆◆◆◆◆◆◆
1700 PRINT "DECISION";\INPUT A$
1720 IF A$<>"CC"GO TO 1740 \GOSUB 4500
1740 IF A$<>"CD"GO TO 1760 \GOSUB 1980
1760 IF A$<>"S"GO TO 1780 \GOSUB 5380
1780 IF A$<>"CS"GO TO 1800 \GOSUB 9940
1800 IF A$<>"PS"GO TO 1840 \GOSUB 10440
1840 IF A$<>"PT"GO TO 1860 \GOSUB 11600
1860 IF A$<>"DT"GO TO 1880 \GOSUB 12040
1880 IF A$<>"SV"GO TO 1900 \GOSUB 12540
1900 IF A$<>"RPT"GO TO 1920 \GOSUB 12880
1920 IF A$<>"HC"GO TO 1940 \GOSUB 13340
1940 IF A$<>"T"GO TO 1640
1960 END
1980 REM ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
2000 REM ◆ CHANGE OF DATA ◆
2020 REM ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
2040 PRINT "DO WANT TO CHANGE (A) PLANNING HORIZON, (B) RESOURCE DATA,"
2060 PRINT "(C) PROJECT DATA";\INPUT A$
2080 IF A$="A"GO TO 2180
2100 IF A$="B"GO TO 2460
2120 IF A$="C"GO TO 2800
2140 GO TO 2040
2160 REM
2180 REM ◆◆◆ PLANNING HORIZON ◆◆◆
2200 REM
2210 IF I<>2GO TO 2220 \PRINT "TREE FROZEN - NOT ALLOWED ";
2212 PRINT "TO CHANGE PLANNING HORIZON"\GO TO 4440
2220 PRINT "NEW PLANNING HORIZON";\INPUT T9
2240 IF T9<=TGO TO 2440
2260 FOR R0=1 TO R
2280 T1=T+1
2300 PRINT "GIVE AMOUNT OF RESOURCE ";TRM$(R$(R0));
2320 PRINT " STARTING AT TIME";T1;\INPUT R2
2340 PRINT "DURATION";\INPUT D1
2360 FOR T0=T1 TO T1+D1-1\R0(T0,R0)=R2\R(T0,R0)=R2\NEXT T0
2380 IF T1+D1-1>=T9GO TO 2420
2400 T1=T0+1\GO TO 2300
2420 NEXT R0
```

```
2440 T=T9\GO TO 4440
2460 REM
2480 REM *** RESOURCE MATRIX ***
2500 REM
2520 PRINT "NAME OF RESOURCE";\INPUT A$
2540 FOR R0=1 TO R\IF A$=R$(R0)GO TO 2580 \NEXT R0
2560 PRINT "WRONG NAME"\GO TO 2520
2580 PRINT "TIME";\INPUT T9
2600 IF T9>=T5GO TO 2640
2620 PRINT "INVALID TIME"\GO TO 2580
2640 PRINT "DURATION";\INPUT D9
2660 PRINT "NEW AMOUNT";\INPUT R9
2680 FOR T0=T9 TO T9+D9-1\R0(T0,R0)=R9\R(T0,R0)=R9\NEXT T0
2700 PRINT "MORE CHANGES OF RESOURCE ";TRM$(R$(R0));\INPUT A$
2720 IF SEG$(A$,1,1)="Y"GO TO 2580
2740 PRINT "DO YOU WANT TO CHANGE ANY OTHER RESOURCE";\INPUT A$
2760 IF SEG$(A$,1,1)="Y"GO TO 2520
2780 GO TO 4440
2800 REM
2820 REM *** PROJECT MATRIX ***
2840 REM
2860 PRINT "DO YOU WANT TO (A) CHANGE, (B) ADD, (C) CANCEL PROJECT";
2880 INPUT A$
2900 IF A$="A"GO TO 2980
2920 IF A$="B"GO TO 3640
2940 IF A$="C"GO TO 4040
2960 GO TO 2860
2980 REM
3000 REM *** CHANGE PROJECT ***
3020 REM
3040 PRINT "NAME OF PROJECT";\INPUT P$
3060 FOR P0=1 TO P\IF P$=P$(P0)GO TO 3090 \NEXT P0
3080 PRINT "WRONG NAME"\GO TO 3040
3090 IF Y=0GO TO 3100 \IF L(P0)>M1GO TO 3100
3095 PRINT "FROZEN TREE - NOT ALLOWED CHANGE"\GO TO 4400
3100 PRINT "CHANGE (A) DESIRED START TIME, (B) RESOURCE REQUEST,"
3120 PRINT "          (C) TREE WIDTH";\INPUT A$
3140 IF A$="A"GO TO 3220
3160 IF A$="B"GO TO 3280
3180 IF A$="C"GO TO 3500
3200 GO TO 3580
3220 PRINT "DESIRED START TIME";\INPUT S0
3240 IF S0>=T5GO TO 3270
3260 PRINT "INVALID TIME"\GO TO 3220
3270 S1=S0-S0(P0)\W0(P0)=W0(P0)+S1\W1(P0)=W1(P0)-S1
3275 S0(P0)=S0\GO TO 3540
3280 PRINT "RESOURCE";\INPUT R$
3300 IF R$="XX"GO TO 3580
3320 PRINT "TIME";\INPUT T9
3340 IF T9>=T5GO TO 3380
3360 PRINT "INVALID TIME"\GO TO 3320
3380 PRINT "DURATION";\INPUT D9
3400 PRINT "AMOUNT";\INPUT P9
3420 FOR R0=1 TO R\IF R$=R$(R0)GO TO 3460 \NEXT R0
3440 PRINT "WRONG NAME OF RESOURCE"\GO TO 3280
3460 FOR T0=T9 TO T9+D9-1\P(T0+K(P0),R0)=P9\NEXT T0
3480 GO TO 3540
3500 PRINT "TREE WIDTH - LEFT";\INPUT W0(P0)
3520 PRINT "TREE WIDTH - RIGHT";\INPUT W1(P0)
3530 W(P0)=W0(P0)+W1(P0)+1
```

```
3540 PRINT "MORE CHANGES FOR PROJECT ";TRM$(P$(P0));\INPUT A$
3560 IF SEG$(A$,1,1)="Y"GO TO 3100
3580 PRINT "DO YOU WANT TO CHANGE ANY OTHER PROJECT";
3600 INPUT A$\IF SEG$(A$,1,1)="Y"GO TO 3040
3620 GO TO 4400
3640 REM
3660 REM ◆◆◆ ADD PROJECT ◆◆◆
3680 REM
3690 IF I<>2GO TO 3700
3695 PRINT "FROZEN TREE - PROJECTS NOT ALLOWED TO BE ADDED"\GO TO 4400
3700 PRINT "NAME OF PROJECT";\INPUT P$
3720 IF P$="XX"GO TO 3980
3740 P=P+1\P$(P)=P$
3760 PRINT "DURATION";\INPUT D(P)
3780 PRINT "DESIRED START TIME";\INPUT S0(P)
3800 IF S0(P)>=T5GO TO 3840
3820 PRINT "INVALID TIME"\GO TO 3780
3840 PRINT "TREE WIDTH - LEFT";\INPUT W0(P)
3845 PRINT "TREE WIDTH - RIGHT";\INPUT W1(P)
3850 W(P)=W0(P)+W1(P)+1
3860 K(P)=K(P-1)+D(P-1)
3880 FOR R0=1 TO R
3900 FOR D0=0 TO D(P)-1\D1=D0+K(P)+1
3920 PRINT "REQUEST ON RESOURCE ";TRM$(R$(R0));" AT PROJECT TIME";D0+1;
3940 INPUT P(D1,R0)\NEXT D0
3960 NEXT R0
3980 PRINT "DO YOU WANT TO ADD MORE PROJECTS";\INPUT A$
4000 IF SEG$(A$,1,1)="Y"GO TO 3700
4020 GO TO 4400
4040 REM
4060 REM ◆◆◆ CANCEL PROJECT ◆◆◆
4080 REM
4090 IF I<>2GO TO 4100
4095 PRINT "FROZEN TREE - PROJECTS NOT ALLOWED ";
4097 PRITN "TO BE CANCELLED"\GO TO 4400
4100 PRINT "NAME OF PROJECT";\INPUT P$
4120 IF P$="XX"GO TO 4360
4140 FOR P0=1 TO P\IF P$=P$(P0)GO TO 4180 \NEXT P0
4160 PRINT "WRONG NAME"\GO TO 4100
4180 FOR P1=P0 TO P-1
4200 K(P1)=K(P1+1)-D(P0)
4220 FOR R0=1 TO R
4240 FOR D0=1 TO D(P1+1)\D1=D0+K(P1)
4260 P(D1,R0)=P(D1+D(P0),R0)
4280 NEXT D0\NEXT R0
4300 P$(P1)=P$(P1+1)\D(P1)=D(P1+1)\S0(P1)=S0(P1+1)
4320 W(P1)=W(P1+1)
4340 NEXT P1\P=P-1
4360 PRINT "DO YOU WANT TO CANCEL MORE PROJECTS";\INPUT A$
4380 IF SEG$(A$,1,1)="Y"GO TO 4040
4400 PRINT "MORE CHANGES OF PROJECT DATA";\INPUT A$
4420 IF SEG$(A$,1,1)="Y"GO TO 2820
4440 PRINT "MORE CHANGES OF DATA";\INPUT A$
4460 IF SEG$(A$,1,1)="Y"GO TO 1980
4480 RETURN
4500 REM ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
4520 REM ◆ CHANGE CONSTRAINTS ◆
4540 REM ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
4600 PRINT "NAME OF RESOURCE";\INPUT A$
4620 FOR R0=1 TO R\IF A$=R$(R0)GO TO 4660 \NEXT R0
```

```
4640 PRINT "WRONG NAME"\GO TO 4600
4660 PRINT "START PERIOD";\INPUT T1
4680 PRINT "DURATION";\INPUT D0
4700 PRINT "DO YOU WANT TO MAKE THE CONSTRAINT SOFT";\INPUT A$
4720 IF SEG$(A$,1,1)<>"Y"GO TO 4740 \T6=100\GO TO 4760
4740 T6=0
4760 FOR T0=T1 TO T1+D0-1\B0(T0,R0)=T6\NEXT T0
4780 PRINT "MORE CHANGED CONSTRAINTS FOR RESOURCE ";
4785 PRINT TRM$(R$(R0));\INPUT A$
4800 IF SEG$(A$,1,1)="Y"GO TO 4660
4820 PRINT "ANY OTHER RESOURCE WITH CHANGED CONSTRAINTS";\INPUT A$
4840 IF SEG$(A$,1,1)="Y"GO TO 4600 \IF SEG$(A$,1,1)<>"N"GO TO 4820
4860 FOR R0=1 TO R
4880 FOR T0=1 TO T\IF B0(T0,R0)<>0GO TO 4920 \NEXT T0
4900 NEXT R0\GO TO 4940
4920 X=1
4940 RETURN
4960 REM ********************
4980 REM * REORDER PROJECTS *
5000 REM ********************
5020 PRINT "NAME OF PROJECT";\INPUT P$
5040 IF P$<>"XX"GO TO 5060 \RETURN
5060 FOR P0=1 TO P\IF P$(L(P0))=P$GO TO 5100 \NEXT P0
5080 PRINT "WRONG NAME"\GO TO 5020
5100 PRINT "POSITION 1 -";P;\INPUT P6
5120 L0=L(P0)
5160 IF P6>P0GO TO 5240
5180 FOR I0=P0 TO P6+1 STEP -1\L(I0)=L(I0-1)
5200 NEXT I0
5220 GO TO 5280
5240 FOR I0=P0 TO P6-1\L(I0)=L(I0+1)
5260 NEXT I0
5280 L(P6)=L0
5300 PRINT "MORE PROJECTS TO REORDER";\INPUT A$
5320 IF SEG$(A$,1,1)="Y"GO TO 5020
5340 IF SEG$(A$,1,1)<>"N"GO TO 5300
5360 RETURN
5380 REM ************
5400 REM * SCHEDULE *
5420 REM ************
5440 REM
5460 REM *** BRANCH THE TREE ***
5480 REM
5490 I=0
5500 W2=0\Y=1
5520 FOR P0=1 TO P
5540 W2(P0)=W0(P0)
5560 IF W1(P0)<=W2(P0)GO TO 5600 \W2(P0)=W1(P0)
5600 IF W(P0)<=W2GO TO 5640
5620 W2=W(P0)
5640 T(P0,1)=S0(P0)
5660 IF W(P0)=-1GO TO 5880
5670 W3(P0)=W(P0)-1-W2(P0)
5680 FOR W0=1 TO W2(P0)
5700 IF W3(P0)<W0GO TO 5844
5720 T(P0,2*W0)=S0(P0)+W0
5760 T(P0,2*W0+1)=S0(P0)-W0
5780 IF T(P0,2*W0+1)>T5GO TO 5820
5800 T(P0,2*W0+1)=T5
5820 IF T(P0,2*W0+1)<=TGO TO 5860
```

```
5840 T(P0,2*W0+1)=T
5842 GO TO 5860
5844 IF W0(P0)<W1(P0)GO TO 5852
5846 T(P0,W1(P0)+1+W0)=S0(P0)-W0
5848 IF S0(P0)-W0>=T5GO TO 5860
5850 T(P0,W1(P0)+1+W0)=T5\GO TO 5860
5852 T(P0,W0(P0)+1+W0)=S0(P0)+W0
5854 IF S0(P0)+W0<=TGO TO 5860
5856 T(P0,W0(P0)+1+W0)=T
5860 NEXT W0
5880 NEXT P0
5900 REM
5920 REM *** ORDER THE PROJECTS ***
5940 REM
5950 IF J=2GO TO 6170
5960 P1=0
5980 FOR W0=-1 TO W2
6000 FOR P0=1 TO P
6020 IF W(P0)<>W0GO TO 6080
6040 P1=P1+1
6060 L(P1)=P0
6080 NEXT P0
6100 NEXT W0
6120 REM
6140 REM *** INITIALIZATION ***
6160 REM
6165 FOR P1=1 TO P\S(P1)=0\NEXT P1
6170 IF J<>0GO TO 6200
6180 FOR P1=1 TO P\S1(P1)=1\NEXT P1
6200 FOR R0=1 TO R\FOR T0=1 TO T
6220 R(T0,R0)=R0(T0,R0)
6240 R1(T0,R0)=R0(T0,R0)
6260 B1(T0,R0)=B0(T0,R0)
6280 NEXT T0\NEXT R0
6300 P2=1
6320 B0=0\B9=0
6340 REM
6360 REM *** SEARCH ***
6380 REM
6400 J=0
6420 FOR P0=P2 TO P\P1=L(P0)
6440 IF W(P1)<>-1GO TO 6460 \S9=1\GO TO 6480
6460 S9=W(P1)
6480 D9=D(P1)-1\K1=K(P1)+1
6500 FOR S0=S1(P1) TO S9
6520 FOR D0=0 TO D9
6540 D1=T(P1,S0)+D0\K2=K1+D0
6560 FOR R0=1 TO R
6580 IF R(D1,R0)+B1(D1,R0)<P(K2,R0)GO TO 6660
6600 NEXT R0
6620 NEXT D0
6640 GO TO 6900
6660 NEXT S0
6665 IF B8<>1GO TO 6680
6670 IF P0-1>MGO TO 6675 \GO TO 6680
6675 B8=0\GO TO 6700
6680 IF B0<B9GO TO 7400
6700 M=P0-1
6720 PRINT
6740 PRINT "YOU ARE AT LEVEL";M
```

```
6760 PRINT "DO YOU WANT TO INTERACT";\INPUT A$
6780 IF SEG$(A$,1,1)<>"N"GO TO 6820
6800 GO TO 6840
6820 IF SEG$(A$,1,1)<>"Y"GO TO 6760 \GOSUB 8440
6840 IF J<>0GO TO 5500 \GO TO 7200
6860 REM
6880 REM *** ASSIGN ***
6900 REM
6920 S(P1)=T(P1,S0)
6940 S1(P1)=S0
6960 FOR D0=0 TO D9
6980 D1=S(P1)+D0\K2=K1+D0
7000 FOR R0=1 TO R
7020 R(D1,R0)=R(D1,R0)-P(K2,R0)
7040 NEXT R0
7060 NEXT D0
7080 PRINT P0;
7100 NEXT P0
7120 IF X<>0GO TO 7820
7140 PRINT \PRINT "AN OPTIMAL SCHEDULE FOUND"\RETURN
7160 REM
7180 REM *** BACKTRACK ***
7200 REM
7220 PRINT "DO YOU WANT TO BACKTRACK";\INPUT A$
7240 IF SEG$(A$,1,1)="Y"GO TO 7360
7260 IF SEG$(A$,1,1)<>"N"GO TO 7220
7280 IF X<>1GO TO 1640
7300 FOR R0=1 TO R\FOR T0=1 TO T\R(T0,R0)=P1(T0,R0)
7320 NEXT T0\NEXT R0\X=0
7340 FOR P0=1 TO P\P1=L(P0)\S(P1)=S2(P1)\NEXT P0\GO TO 1640
7360 PRINT "HOW MANY BACKTRACKS BEFORE NEXT STOP";\INPUT B9
7380 B0=0
7385 PRINT "DO YOU WANT TO STOP WHEN A LOWER TREE LEVEL IS
                         REACHED";\INPUT A$
7390 IF A$<>"Y"GO TO 7400 \B8=1
7400 IF P0=P3GO TO 7420 \GO TO 7440
7420 S=1\GO TO 7740
7440 B0=B0+1
7460 S1(P1)=1\S(P1)=0
7480 P0=P0-1\P2=P0\P1=L(P0)
7500 IF I<>2GO TO 7560 \IF P0<>M1GO TO 7560
7520 PRINT "YOU ARE BACK AT LEVEL";M1
7540 GO TO 9360
7560 D9=D(P1)-1\K1=K(P1)+1
7580 FOR D0=0 TO D9\D1=S(P1)+D0\K2=K1+D0
7600 FOR R0=1 TO R
7620 R(D1,R0)=R(D1,R0)+P(K2,R0)
7640 NEXT R0
7660 NEXT D0
7680 S1(P1)=S1(P1)+1
7700 IF S1(P1)>W(P1)GO TO 6680
7720 IF S<>1GO TO 6420
7740 S=0\S(L(1))=0
7760 PRINT "NO OPTIMAL SCHEDULE FOUND"\RETURN
7780 REM
7800 REM *** TEST OF OPTIMALITY ***
7820 REM
7840 B1=0
7860 FOR R0=1 TO R\B1(R0)=0
7880 FOR T0=1 TO T
```

```
7900 IF R(T0,R0)>=B1(R0)GO TO 7940
7920 B1(R0)=R(T0,R0)
7940 NEXT T0
7960 IF B1(R0)>=B1GO TO 7980 \B1=B1(R0)
7980 NEXT R0
8020 IF B1<>0GO TO 8140
8060 PRINT
8080 PRINT "AN OPTIMAL SCHEDULE FOUND"\X=0\RETURN
8100 REM
8120 REM *** SAVE IMPROVED SCHEDULE ***
8140 REM
8160 PRINT
8180 PRINT "AN IMPROVED SCHEDULE FOUND"
8220 FOR R0=1 TO R
8240 PRINT "MAX OVERLOAD FOR RESOURCE ";TRM$(R$(R0));" : ";-B1(R0)
8260 FOR T0=1 TO T
8280 R1(T0,R0)=R(T0,R0)
8300 IF B0(T0,R0)=0GO TO 8340
8320 B1(T0,R0)=-B1-1
8340 NEXT T0
8360 NEXT R0
8380 FOR P0=1 TO P\P1=L(P0)\S2(P1)=S(P1)\NEXT P0
8400 P1=P+1\P0=P+1
8420 GO TO 6680
8440 REM *************
8460 REM * INTERACT *
8480 REM *************
8500 P4=P0\P2=P0
8520 PRINT "INTERACTION DECISION";\INPUT A$
8540 IF A$<>"DT"GO TO 8580 \GOSUB 12040
8580 IF A$<>"PS"GO TO 8620 \GOSUB 10440
8600 IF A$<>"RS"GO TO 8620 \GOSUB 11000
8620 IF A$<>"PT"GO TO 8640 \GOSUB 11600
8640 IF A$<>"CS"GO TO 8660 \GOSUB 9940
8660 IF A$<>"HC"GO TO 8670 \GOSUB 13340
8670 IF A$<>"RPT"GO TO 8680 \GOSUB 12880
8680 IF A$<>"FR"GO TO 9220
8720 REM
8740 REM *** FREEZE ***
8760 REM
8770 B8=1
8780 OPEN "FROZ" FOR OUTPUT AS FILE VF3$(1000)
8800 VF3(1)=STR$(T)\VF3(2)=STR$(P)
8820 FOR R0=1 TO R\V0=(R0-1)*100+10
8840 FOR T0=1 TO T\T1=V0+1+(T0-1)*2
8860 VF3(T1)=STR$(R0(T0,R0))
8880 VF3(T1+1)=STR$(B0(T0,R0))
8900 NEXT T0
8920 NEXT R0
8940 FOR P0=1 TO P\V0=400+P0*50
8960 VF3(V0)=P$(P0)
8980 VF3(V0+1)=STR$(S0(P0))
9000 VF3(V0+2)=STR$(W0(P0))
9020 VF3(V0+3)=STR$(W1(P0))
9040 VF3(V0+4)=STR$(L(P0))
9050 VF3(V0+5)=STR$(S1(P0))
9060 V0=V0+6
9080 FOR R0=1 TO R
9100 FOR D0=1 TO D(R0)\D1=D0+K(R0)
9110 V0=V0+1
```

```
9120 VF3(V0)=STR$(P(D1,R0))
9140 NEXT D0
9160 NEXT R0
9180 NEXT P0
9190 M1=M
9200 I=2
9220 IF A$<>"UFR"GO TO 9240 \GO TO 9360
9240 IF A$<>"CD"GO TO 9260 \GOSUB 1980 \J=1
9260 IF A$<>"PO"GO TO 9300 \GOSUB 4960 \J=2
9262 IF A$<>"CAT"GO TO 9300
9300 IF A$<>"C"GO TO 8520
9320 P0=P4\P1=L(P0)
9340 RETURN
9360 REM ◆◆◆◆◆◆◆◆◆◆◆◆
9400 REM ◆ UNFREEZE ◆
9420 REM ◆◆◆◆◆◆◆◆◆◆◆◆
9440 I=0\J=0
9460 M1=0
9480 T=VAL(VF3(1))\P=VAL(VF3(2))
9500 FOR R0=1 TO R\V0=(R0-1)◆100+10
9520 FOR T0=1 TO T\T1=V0+1+(T0-1)◆2
9540 R0(T0,R0)=VAL(VF3(T1))
9560 B0(T0,R0)=VAL(VF3(T1+1))
9580 NEXT T0
9600 NEXT R0
9620 D(0)=0\K(0)=0
9640 FOR P0=1 TO P\V0=400+P0◆50
9660 P$(P0)=VF3(V0)
9680 S0(P0)=VAL(VF3(V0+1))
9700 W0(P0)=VAL(VF3(V0+2))
9720 W1(P0)=VAL(VF3(V0+3))
9740 L(P0)=VAL(VF3(V0+4))
9750 S1(P0)=VAL(VF3(V0+5))
9760 V0=V0+6\K(P0)=K(P0-1)+D(P0-1)
9780 FOR R0=1 TO R
9800 FOR D0=1 TO D(P0)\D1=D0+K(P0)
9820 V0=V0+1
9840 P(D1,R0)=VAL(VF3(V0))
9860 NEXT D0
9880 NEXT R0
9900 NEXT P0
9910 CLOSE VF3
9920 GO TO 6200
9940 REM ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
9960 REM ◆ COMPLETION OF SCHEDULE ◆
9980 REM ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
10000 FOR R0=1 TO R\FOR T0=1 TO T\R1(T0,R0)=R(T0,R0)\NEXT T0\NEXT R0
10020 FOR P0=P2 TO P\IF Y<>0GO TO 10040 \P1=P0\GO TO 10060
10040 P1=L(P0)
10060 S(P1)=S0(P1)
10080 FOR R0=1 TO R
10100 FOR D0=0 TO D(P1)-1\D1=D0+S(P1)
10120 R(D1,R0)=R(D1,R0)-P(K(P1)+1+D0,R0)
10140 NEXT D0\NEXT R0\NEXT P0
10160 PRINT "OPTIONS (PS),(PT),(HC),(RPT),(SV),(C)";\INPUT A$
10180 IF A$<>"PS"GO TO 10220 \GOSUB 10440
10220 IF A$<>"PT"GO TO 10240 \GOSUB 11600
10240 IF A$<>"HC"GO TO 10260 \GOSUB 13340
10260 IF A$<>"SV"GO TO 10270 \GOSUB 12540
10270 IF A$<>"RPT"GO TO 10280 \GOSUB 12880
```

```
10280 IF A$<>"C"GO TO 10160
10300 A$="XZ"
10320 FOR R0=1 TO R
10340 FOR T0=1 TO T
10360 R(T0,R0)=R1(T0,R0)
10380 NEXT T0\NEXT R0
10390 FOR P0=1 TO P\IF Y=0GO TO 10410
10400 P1=L(P0)
10410 S(P1)=0
10415 NEXT P0
10420 RETURN
10440 REM ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
10460 REM ◆ PROJECT SCHEDULE ◆
10480 REM ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
10500 PRINT CHR$(1)\PRINT CHR$(12)
10540 PRINT #F:"PROJECT SCHEDULE"
10560 FOR C0=1 TO 60\PRINT #F:"-";\NEXT C0
10580 PRINT #F:
10600 PRINT #F:TAB(8);":";TAB(30);"TIME"
10620 A0$="PROJECT :          1    1    2    2    3    3    4    4    5"
10640 PRINT #F:A0$
10660 A0$="        : 1   5    0    5    0    5    0    5    0    5    0"
10680 PRINT #F:A0$
10700 FOR C0=1 TO 60\PRINT #F:"-";\NEXT C0
10720 PRINT #F:
10740 FOR P0=1 TO P\IF Y<>0GO TO 10780
10760 P1=P0\GO TO 10800
10780 P1=L(P0)
10800 PRINT #F:TAB(3);TRM$(P$(P1));TAB(8);":";
10820 IF X<>1GO TO 10840 \C1=S2(P1)+10\GO TO 10860
10840 C1=S(P1)+10
10860 PRINT #F:TAB(C1-1);
10880 FOR C0=C1 TO C1+D(P1)-1\PRINT #F:"◆";\NEXT C0
10900 PRINT #F:
10920 NEXT P0
10940 FOR C0=1 TO 60\PRINT #F:"-";\NEXT C0
10960 PRINT #F:
11160 PRINT #F:"RESOURCE:"
11220 FOR C0=1 TO 60\PRINT #F:"-";\NEXT C0\PRINT #F:
11240 FOR R0=1 TO R
11260 PRINT #F:TAB(2);STR$(R0);" CAP : ";
11280 FOR T0=1 TO T
11320 PRINT #F:STR$(R0(T0,R0));
11340 NEXT T0\PRINT #F:
11360 PRINT #F:TAB(4);"UTIL: ";
11380 FOR T0=1 TO T
11400 IF X<>1GO TO 11420 \R2=R0(T0,R0)-R1(T0,R0)\GO TO 11440
11420 R2=R0(T0,R0)-R(T0,R0)
11440 IF R2>9GO TO 11500
11460 PRINT #F:STR$(R2);
11480 GO TO 11520
11500 PRINT #F:"◆";
11520 NEXT T0\PRINT #F:
11540 NEXT R0
11560 FOR C0=1 TO 60\PRINT #F:"-";\NEXT C0\PRINT #F:
11580 RETURN
11600 REM ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
11620 REM ◆ PROJECT TABLE ◆
11640 REM ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
11660 PRINT CHR$(1)\PRINT CHR$(12)
```

```
11680 PRINT #F:"PROJECT TABLE"
11700 FOR B0=1 TO 60\PRINT #F:"-";\NEXT B0\PRINT #F:
11720 PRINT #F:"PROJ  DUR       START TIME                   RES  USAGE"
11730 PRINT #F:"              EARLY DES  LATE ASS"
11740 FOR B0=1 TO 60\PRINT #F:"-";\NEXT B0\PRINT #F:
11760 FOR P0=1 TO P\IF Y<>0GO TO 11800
11780 P1=P0\GO TO 11810
11800 P1=L(P0)
11810 S0=S0(P1)-W0(P1)\IF S0>0GO TO 11815 \S0=1
11815 S1=S0(P1)+W1(P1)\IF S1<=TGO TO 11820 \S1=T
11820 PRINT #F:TRM$(P$(P1));TAB(5);D(P1);TAB(11);STR$(S0);
11830 PRINT #F:TAB(16);S0(P1);TAB(21);
11835 PRINT #F:S1;TAB(26);
11840 IF X<>1GO TO 11860 \PRINT #F:S2(P1);\GO TO 11900
11860 PRINT #F:S(P1);
11900 FOR R0=1 TO R\PRINT #F:TAB(40);TRM$(R$(R0));TAB(45);
11920 FOR D0=1 TO D(P1)
11940 PRINT #F:STR$(R(K(P1)+D0,R0));
11960 NEXT D0\PRINT #F:
11980 NEXT R0\NEXT P0
12000 FOR B0=1 TO 60\PRINT #F:"-";\NEXT B0\PRINT #F:
12020 RETURN
12040 REM ******************
12060 REM * DECISION TABLE *
12080 REM ******************
12100 PRINT "DO YOU WANT A HARD COPY";\INPUT A$
12120 IF SEG$(A$,1,1)<>"Y"GO TO 12160
12140 OPEN "KB:" FOR OUTPUT AS FILE #6\F=6
12160 PRINT #F:"FOLLOWING DECISIONS ARE AVAILABLE"\PRINT #F:
12180 PRINT #F:"C   = CONTINUE"
12200 PRINT #F:"CC  = CHANGE OF SOFT CONSTRAINT"
12220 PRINT #F:"CD  = CHANGE DATA".
12240 PRINT #F:"CS  = COMPLETION OF SCHEDULE"
12260 PRINT #F:"DT  = DECISION TABLE"
12280 PRINT #F:"FR  = FREEZE TREE"
12300 PRINT #F:"HC  = HARD COPY"
12320 PRINT #F:"PS  = PROJECT SCHEDULE"
12340 PRINT #F:"PT  = PROJECT TABLE"
12360 PRINT #F:"RO  = REORDER PROJECTS"
12380 PRINT #F:"RPT= REPORT"
12420 PRINT #F:"S   = SCHEDULE"
12440 PRINT #F:"SV  = SAVE SCHEDULE"
12460 PRINT #F:"T   = TERMINATE"
12480 PRINT #F:"UFR= UNFREEZE TREE"
12500 IF F<>6GO TO 12520 \CLOSE #6
12520 F=0\RETURN
12540 REM *****************
12560 REM * SAVE SCHEDULE *
12580 REM *****************
12600 PRINT "NAME OF OUTPUT FILE";\INPUT O$
12620 OPEN O$ FOR OUTPUT AS FILE VF2$(1000)
12640 VF2(1)=STR$(T)\VF2(2)=STR$(R)\VF2(3)=STR$(P)
12660 FOR R0=1 TO R\V0=R0*50\VF2(V0)=R$(R0)
12680 FOR T0=1 TO T\VF2(V0+T0)=STR$(R0(T0,R0))\NEXT T0\NEXT R0
12700 FOR P0=1 TO P\V0=200+P0*50
12720 VF2(V0)=P$(P0)\VF2(V0+1)=STR$(D(P0))
12740 IF Y<>0GO TO 12750 \S(P0)=S0(P0)
12750 S0=S(P0)-S0(P0)\S1=W0(P0)+S0\IF S1>0GO TO 12755 \S1=0
12755 S2=W1(P0)-S0\IF S2>0GO TO 12760 \S2=0
12760 VF2(V0+2)=STR$(S(P0))\VF2(V0+3)=STR$(S1)
```

```
12770 VF2(V0+4)=STR$(S2)
12780 V0=V0+5
12800 FOR P0=1 TO R\FOR D0=1 TO D(P0)\D1=D0+K(P0)
12820 V0=V0+1\VF2(V0)=STR$(P(D1,P0))
12840 NEXT D0\NEXT P0\NEXT P0\CLOSE VF2
12860 RETURN
12880 REM **********
12900 REM * REPORT *
12920 REM **********
12940 OPEN "KB:" FOR OUTPUT AS FILE #7
12960 FOR B0=1 TO 72\PRINT #7:"-";\NEXT B0\PRINT #7:\PRINT #7:
12980 PRINT "NAME OF JOB";\INPUT A$
13000 PRINT #7:A$;TAB(60);DAT$
13020 PRINT #7:"INPUT FILE: ";F$
13080 FOR C0=1 TO 3\PRINT #7\NEXT C0
13100 CLOSE #7\CALL "SETC"(3)\CALL "WAIT"(1)
13120 OPEN "KB:" FOR OUTPUT AS FILE #7
13140 F=7\GOSUB 10440
13200 PRINT #F\PRINT #F
13220 GOSUB 11600
13240 FOR N0=24+3*P+2*R TO 66\PRINT #F\NEXT N0
13260 FOR C0=1 TO 72\PRINT #7:"-";\NEXT C0
13280 FOR C0=1 TO 7\PRINT #7\NEXT C0
13300 CLOSE #7\CALL "SETC"(3)\CALL "WAIT"(1)
13320 F=0\RETURN
13340 REM *************
13360 REM * HARD COPY *
13380 REM *************
13400 F=7
13420 OPEN "KB:" FOR OUTPUT AS FILE #7
13440 PRINT "ID";\INPUT A$
13460 FOR I=1 TO 3\PRINT #7\NEXT I
13480 FOR C0=1 TO 72\PRINT #7:"-";\NEXT C0\PRINT #7\PRINT #7
13500 PRINT #7:A$;TAB(60);DAT$
13510 PRINT #7:"INPUT FILE: ";F$
13520 FOR C0=1 TO 3\PRINT #7\NEXT C0
13540 CLOSE #7\CALL "SETC"(3)\CALL "WAIT"(1)
13560 OPEN "KB:" FOR OUTPUT AS FILE #7
13580 PRINT "(PS),(PT) OR (C)";\INPUT A$
13600 IF A$<>"PS"GO TO 13640 \GOSUB 10440
13640 IF A$<>"PT"GO TO 13660 \GOSUB 11600
13660 FOR C0=1 TO 4\PRINT #7\NEXT C0
13680 IF A$<>"C"GO TO 13580
13690 A$="XZ"
13700 FOR B0=1 TO 72\PRINT #7:"-";\NEXT B0
13720 FOR I=1 TO 7\PRINT #7\NEXT I
13740 CLOSE #7\CALL "SETC"(3)\CALL "WAIT"(1)
13760 F=0\RETURN
```

APPENDIX II

LIST OF VARIABLES IN PROGRAM AND THEIR FUNCTION

Variable	Function
MATRICES	
RØ(TØ,RØ)	Initially available amount of resource RØ in time TØ
R(TØ,RØ)	Remaining amount of resource RØ in time TØ
R1(TØ,RØ)	Remaining amount of resource RØ in time TØ for saved schedule
BØ(TØ,RØ)	Initial bounding value for overloading of resource RØ in time TØ
B1(TØ,RØ)	Actual bonding value for overloading of resource RØ in time TØ
P(DØ,RØ)	Requested amount of resource RØ by a project at time DØ
T(PØ,WØ)	Implied start time for project PØ when branch WØ is used
VECTORS	
D(PØ)	Duration of project PØ
K(PØ)	Points to project PØ in the request matrix
SØ(PØ)	Desired start time for project PØ
S(PØ)	Assigned start time for project PØ
S2(PØ)	Assigned start time for project PØ in saved schedule
WØ(PØ)	Tree width-left for project PØ
W1(PØ)	Tree width-right for project PØ

Variable	Function
W(PØ)	Tree width-total for project PØ
W2(PØ)	Max. of WØ(PØ) and W1(PØ) for project PØ
W3(PØ)	Min. of WØ(PØ) and W1(PØ) for project PØ
L(PØ)	List of ordered projects
S1(PØ)	Boundary index for tree branch for project PØ
B1(RØ)	Maximum overload of resource RØ
STRINGS	
R$(RØ)	Name of resource RØ
P$(PØ)	Name of project PØ
F$	Name of input file
O$	Name of output file
A$, P$, R$	Used for input from video terminal
UNSUBSCRIPTED VARIABLES	
T	Planning horizon
R	No. of resources
P	No. of projects
RØ	Index for resources
VØ	Index for virtual files
TØ	Index for schedule time
PØ,P1	Index for projects
DØ,D1	Index for project time
T5	Start time for updating
P3	Pointer for tree root
X	Constraint switch; 1 if soft constraint; Ø otherwise
P2	Boundary index for tree level

Variable	Function
F	File number
Y	Schedule transformer, 1 if scheduling not started; Ø otherwise
B1	Max overload
W2	Max tree width
I	Interact switch; 2 if tree frozen; Ø otherwise
WØ	Index for branches
BØ	Backtrack counter
B9	Limit for backtrack counter
J	Interact switch; 1 if data is changed; 2 if projects are reordered; Ø otherwise
M	Level for last assigned project
M1	Pointer for root of frozen tree

APPENDIX III

DESCRIPTION OF PROGRAM ELEMENTS

Input of Data

Reads data from the specified file. Planning horizon [T], number of resources [R] and number of projects [P] are first read. After this, the available amount in each time unit is read for each resource [RØ(TØ,RØ)]. For each project, name [P$(PØ)], duration [D(PØ)], desired start time [SØ(PØ)], tree width [WØ(PØ),W1(PØ)] and request in each time unit [P(DØ,RØ)] is input into the program. If updating of a previous schedule is at hand, the start time for the updating [T5] is specified.

Decision

Gives the operator opportunity to change data or constraints, review schedules, get hard copies of selected output, save schedule on specified file, start or terminate the scheduling procedure.

Change Data

Allows the operator to change planning horizon, available amounts of resources, desired start times, tree width and requested amount of each resource. Also, new projects can be added and old ones cancelled.

Change Constraint

Initially all constraints are regarded as hard, i.e., no overload permitted. Arbitrary constraints are changed into soft if so specified. A soft constraint can also be changed back to a hard constraint. A test is performed to check if there are any soft constraints. If so, the switch X is changed from Ø to 1.

Reorder Projects

Initially all projects are ordered with respect to their number of branches. Here the operator can change this order by specifying at what level a certain project is to be fit in.

Schedule

This is the main program element. First the number of branches and their corresponding start times [T(PØ,WØ)] are found. A test is performed to check whether these start times are within the planning horizon. If not, the start times are set either to the beginning or to the end of the planning period. After branching the tree, the projects are ordered in a list [L(PØ)] with respect to their number of branches.

Upon initalization, the search is started. For one project at a time, starting from the root of the tree, available amount of resources are checked against requested amount in each time period of the project duration. If one branch is not possible, the next one is tried until either all requests are satisfied or all branches tried without satisfying the request. In the first case, the project is assigned the start time corresponding to the actual branch and available amounts of resources are reduced. The tree level is printed on the video screen and next project is chosen. If it is possible to assign all projects the solution achieved is tested for optimality, i.e., maximum overload [B1(RØ)] is calculated for each resource and checked if they are equal to zero. If so, an optimal schedule is found and the control returns to DECISION. If not, an improved schedule is found, the overloading matrix is updated and the schedule stored.

When all projects cannot be assigned without backtracking the computer stops and asks for directions. The operator can choose to interact before continuing the search. If so, he calls INTERACT. If not, the operator specifies if he wants to backtrack. If not, the control returns to DECISION. If he wants to backtrack, he specifies a limit for the backtracking [B9] and also specifies whether a stop is wanted when a lower tree level is reached or not [B8]. Then the search continues.

Interact

The operator has the same possibilities as in DECISION to change data and constraints, and review schedules. Before performing these changes, he can save the schedule down to the present level of search by freezing the tree. After this, changes can only be done to that part of the tree below the current level of search. If an optimal schedule cannot be found, the frozen part of the tree is restored by unfreezing the tree. Furthermore, the operator can reorder the projects by calling REORDER PROJECTS.

Completion of Partial Schedule

Completes a partial schedule by assigning projects at their desired start times.

Project Schedule

Creates and prints a barchart showing the assignment for each project and the utilization of each resource in each time period.

Project Table

Creates and prints a table showing pertinent data for each project.

Decision Table

Table of available decisions and their code.

Save Schedule

Saves a schedule on a file for future use as input to the program.

Report, Hard Copy

Prints a hard copy in standardized form (REPORT) or of selected output (HARD COPY).

APPENDIX IV

CREATION OF INPUT FILE

An input data file for the scheduling program is created by calling the program CREFIL. The source listing is given below. The variables used are explained in Appendix II. With this the code is, hopefully, self-explanatory. Before attempting to create a file it is, however, recommended to organize data in the way they are input into the file.

```
CREFIL.BAS          08-AUG-77                                  PAGE 1

100 REM          **************************************
120 REM          * THIS PROGRAM CREATES AN INPUT FILE *
140 REM          *            FOR SCHEDA.BAS          *
160 REM          **************************************
180 DIM R0(40,3)
200 DIM P(100,3)
220 DIM P$(10),D(10),S0(10),W0(10),W1(10)
240 PRINT "NAME OF NEW INPUT FILE";\INPUT F$
260 OPEN F$ FOR OUTPUT AS FILE VF1$(1000)
280 PRINT "PLANNING HORIZON";\INPUT T\VF1(1)=STR$(T)
300 PRINT "NO OF RESOURCES";\INPUT R\VF1(2)=STR$(R)
320 PRINT "NO OF PROJECTS";\INPUT P\VF1(3)=STR$(P)
340 REM
360 REM *** RESOURCE MATRIX ***
380 REM
400 FOR R0=1 TO R
420 V0=R0*50
440 PRINT "NAME OF RESOURCE";R0;
460 INPUT R$(R0)\VF1(V0)=R$(R0)
480 FOR T0=1 TO T
500 PRINT "AVAILABLE AMOUNT OF RESOURCE";R0;"AT TIME";T0;
520 INPUT R0(T0,R0)\VF1(V0+T0)=STR$(R0(T0,R0))
540 NEXT T0
560 NEXT R0
580 REM
600 REM *** PROJECT MATRIX ***
620 REM
640 FOR P0=1 TO P
660 V0=200+P0*50
680 PRINT "FOR PROJECT";P0;"GIVE NAME";\INPUT P$(P0)
700 PRINT "DURATION";\INPUT D(P0)
720 PRINT "DESIRED START TIME";\INPUT S0(P0)
740 PRINT "TREE WIDTH - LEFT";\INPUT W0(P0)
760 PRINT "TREE WIDTH - RIGHT";\INPUT W1(P0)
780 VF1(V0)=P$(P0)
800 VF1(V0+1)=STR$(D(P0))
820 VF1(V0+2)=STR$(S0(P0))
840 VF1(V0+3)=STR$(W0(P0))
860 VF1(V0+4)=STR$(W1(P0))
880 V0=V0+5
900 FOR R0=1 TO R
920 FOR D0=1 TO D(P0)
940 V0=V0+1
960 PRINT "REQUIRED AMOUNT OF RESOURCE ";R0;" AT PROJECT TIME ";D0;
980 INPUT P(D0,R0)\VF1(V0)=STR$(P(D0,R0))
1000 NEXT D0
1020 NEXT R0
1040 NEXT P0
1060 CLOSE VF1
1080 END
```